Splitting Bisimulations and Retrospective Conditions

J.A. Bergstra[a,b], C.A. Middelburg[c,a,*]

[a]*Programming Research Group, University of Amsterdam, P.O. Box 41882, 1009 DB Amsterdam, Netherlands*

[b]*Department of Philosophy, Utrecht University, P.O. Box 80126, 3508 TC Utrecht, Netherlands*

[c]*Computing Science Department, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, Netherlands*

Abstract

We investigate conditional expressions in the setting of ACP, an algebraic theory about processes. We introduce ACP^{c}, an extension of ACP with conditional expressions in which the conditions are taken from a free Boolean algebra over a set of generators, and also its main models, called full splitting bisimilation models. We add two simple mechanisms for condition evaluation to ACP^{c}; and we show their connection with state operators and signal emission, mechanisms from other extensions of ACP usable for condition evaluation. To allow for looking back on conditions under which preceding actions have been performed, we add a retrospection operator on conditions to ACP^{c}. The choice of conditions forces us to introduce a new variant of bisimulation. However, without the generality implied by that choice, it would not have been possible to extend ACP^{c} with retrospection. The addition of retrospection is a basic way to increase expressiveness.

Key words: Splitting bisimulation, Retrospective conditions, Condition evaluation, State operators, Signal emission, Process algebra, Boolean algebras

* Corresponding author.
Email addresses: `janb@science.uva.nl` (J.A. Bergstra), `keesm@win.tue.nl` (C.A. Middelburg).
URLs: `http://www.science.uva.nl/~janb` (J.A. Bergstra), `http://www.win.tue.nl/~keesm` (C.A. Middelburg).

1 Introduction

Many theories about processes include conditional expressions of some form. For instance, several extensions of ACP [1,2] include conditional expressions of the form $\zeta :\rightarrow p$ or $p \lhd \zeta \rhd q$ (see e.g. [3–6]). What are considered to be conditions and how they are evaluated differs from one case to another. This state of affairs forms part of our motivation to investigate this matter further. The set of conditions is usually one of the following:

- a two-valued set, usually called $\mathbb{B}$;
- the set of all propositions with propositional variables from a given set and with finite conjunctions and disjunctions;
- the domain of a free Boolean algebra over a given set of generators.

The last alternative generalizes both other alternatives. In this paper, we will focus our attention on the last alternative and implicate the other alternatives where appropriate for explanation or motivation.

We introduce $\mathrm{ACP}^{\mathrm{c}}$, an extension of ACP with conditional expressions of the form $\zeta :\rightarrow p$ in which the set of conditions is the domain of a free Boolean algebra over a given set of generators. We present the main models of $\mathrm{ACP}^{\mathrm{c}}$, which are based on labelled transition systems of which the labels consist of a condition and an action, called conditional transition systems, and a variant of bisimilarity in which a transition of one of the related transition systems may be simulated by a set of transitions of the other transition system, called splitting bisimilarity.

The presented models of $\mathrm{ACP}^{\mathrm{c}}$ demonstrate that infinitely branching processes can be covered, even in the case where the set of generators of the free Boolean algebra is not restricted to be finite. An infinite set of generators is needed for the extension of $\mathrm{ACP}^{\mathrm{c}}$ with the retrospection operator on conditions mentioned below. The approach of structural operational semantics [7] can only be used here to describe the model that covers only finitely branching processes.

Existing mechanisms that allow for a kind of condition evaluation in conditional expressions include the state operators as introduced in [4] and signal emission as introduced in [6]. However, those mechanisms were not devised for that purpose. We extend $\mathrm{ACP}^{\mathrm{c}}$ with operators devised for condition evaluation, with state operators, and with signal emission; and show how those extensions are related. For the main models of $\mathrm{ACP}^{\mathrm{cs}}$, the extension of $\mathrm{ACP}^{\mathrm{c}}$ with signal emission, generalizations of conditional transition systems and splitting bisimilarity are introduced.

Two kinds of operators are devised for condition evaluation, one for the case where condition evaluation is not dependent on process behaviour and the other for the case where condition evaluation is dependent on process behaviour. We show how a theory about the set of atomic conditions can be used for condition evaluation with an operator of the former kind, that the operators of the former kind are superseded by the operators of the latter kind and that those operators are in their turn superseded by the state operators. We also show that the signal emission operator corresponds to a local form of condition evaluation: unlike the forms of condition evaluation covered by the operators mentioned above, condition evaluation by means of the signal emission operator does not persist over performing an action.

We also extend $\mathrm{ACP}^{\mathrm{c}}$ with a retrospection operator on conditions, which allows for looking back on conditions under which preceding actions have been performed. For the main models of $\mathrm{ACP}^{\mathrm{cr}}$, the extension of $\mathrm{ACP}^{\mathrm{c}}$ with the retrospection operator, an adaptation of splitting bisimilarity is introduced. We extend $\mathrm{ACP}^{\mathrm{cr}}$ with the above-mentioned operators devised for condition evaluation as well.

The addition of retrospection to $\mathrm{ACP}^{\mathrm{c}}$ is a basic way to increase expressiveness. We have not yet formed a clear notion of the applications of this addition. We outline in this paper a process algebra built on $\mathrm{ACP}^{\mathrm{cr}}$ in which retrospection allows for using conditions which express that a certain number of steps ago a certain action must have been performed. This suggests, for example, that we can deal with the history pointers from [8] using retrospection. The effect of retrospection on expressiveness forms part of our motivation to develop $\mathrm{ACP}^{\mathrm{cr}}$.

The work presented in this paper, can easily be adapted to other process algebras based on (strong) bisimulation models, such as the strong bisimulation version of CCS [9]. Adaptation to CSP [10], which is not based on bisimulation models, will be more difficult and in part perhaps even impossible.

The structure of this paper is as follows. First of all, we introduce $\mathrm{BPA}^{\mathrm{c}}_{\delta}$, an important subtheory of $\mathrm{ACP}^{\mathrm{c}}$ that does not support parallelism and communication (Section 2). After that, we introduce conditional transition systems, splitting bisimilarity of conditional transition systems (Section 3) and the full splitting bisimulation models of $\mathrm{BPA}^{\mathrm{c}}_{\delta}$, the main models of $\mathrm{BPA}^{\mathrm{c}}_{\delta}$ (Section 4). Following this, we have a closer look at splitting bisimilarity based on structural operational semantics (Section 5). Next, we extend $\mathrm{BPA}^{\mathrm{c}}_{\delta}$ to $\mathrm{ACP}^{\mathrm{c}}$ (Section 6) and expand the full splitting bisimulation models of $\mathrm{BPA}^{\mathrm{c}}_{\delta}$ to full splitting bisimulation models of $\mathrm{ACP}^{\mathrm{c}}$ (Section 7). Then, we extend $\mathrm{ACP}^{\mathrm{c}}$ with guarded recursion (Section 8). Thereupon, we extend $\mathrm{ACP}^{\mathrm{c}}$ with condition evaluation operators (Section 9), with state operators (Section 10) and with a signal emission operator (Section 11); and analyse how those operators are related. We also adapt the full splitting bisimulation models of $\mathrm{ACP}^{\mathrm{c}}$ to the full signal-observing splitting bisimulation models of $\mathrm{ACP}^{\mathrm{cs}}$, the extension of $\mathrm{ACP}^{\mathrm{c}}$ with signal emission (Section 12). After that, we extend $\mathrm{BPA}_{\delta}^{\mathrm{c}}$ with a retrospection operator (Section 13) and adapt the full splitting bisimulation models of $\mathrm{BPA}_{\delta}^{\mathrm{c}}$ to the full retrospective splitting bisimulation models of $\mathrm{BPA}_{\delta}^{\mathrm{cr}}$, the extension of $\mathrm{BPA}_{\delta}^{\mathrm{c}}$ with retrospection (Section 14). Next, we extend $\mathrm{BPA}_{\delta}^{\mathrm{cr}}$ to $\mathrm{ACP}^{\mathrm{cr}}$ (Section 15) and expand the full retrospective splitting bisimulation models of $\mathrm{BPA}_{\delta}^{\mathrm{cr}}$ to full retrospective splitting bisimulation models of $\mathrm{ACP}^{\mathrm{cr}}$ (Section 16). Thereupon, we extend $\mathrm{ACP}^{\mathrm{cr}}$ with condition evaluation operators as well (Section 17). We also outline an interesting variant of $\mathrm{ACP}^{\mathrm{cr}}$ (Section 18). Finally, we make some remarks about related work and mention some options for future work (Section 19).

Some familiarity with Boolean algebras is desirable. The definitions of all notions concerning Boolean algebras that are used in this paper can, for example, be found in [11].

2 BPA with Conditions

BPA_{δ} is a subtheory of ACP that does not support parallelism and communication (see e.g. [2]). In this section, we present an extension of BPA_{δ} with guarded commands, i.e. conditional expressions of the form $\zeta :\rightarrow p$. The extension is called $\mathrm{BPA}_{\delta}^{\mathrm{c}}$. In the extension, just as in BPA_{δ}, it is assumed that a fixed but arbitrary finite set of *actions* A, with $\delta \notin \mathsf{A}$, has been given. Moreover it is assumed that a fixed but arbitrary set of *atomic conditions* C_{at} has been given.

Let κ be an infinite cardinal. Then $\mathcal{C}_{\kappa}$ is the free κ-complete Boolean algebra over C_{at}. [1] As usual, we identify Boolean algebras with their domain. Thus, we also write $\mathcal{C}_{\kappa}$ for the domain of $\mathcal{C}_{\kappa}$. It is well known that, if κ is regular, [2] $\mathcal{C}_{\kappa}$ is isomorphic to the Boolean algebra of equivalence classes with respect to logical equivalence of the set of all propositions with elements of C_{at} as propositional variables and with conjunctions and disjunctions of less than κ propositions (see e.g. [11]). In $\mathrm{BPA}_{\delta}^{\mathrm{c}}$, conditions are taken from $\mathcal{C}_{\aleph_0}$. Moreover, if C_{at} is a finite set, then $\mathcal{C}_{\kappa} = \mathcal{C}_{\aleph_0}$ for all cardinals $\kappa > \aleph_0$. We are also interested in $\mathcal{C}_{\kappa}$ for cardinals $\kappa > \aleph_0$ because it permits us to consider infinitely branching processes in the case where C_{at} is an infinite set. Henceforth, we write $\mathcal{C}$ for $\mathcal{C}_{\aleph_0}$.

The algebraic theory $\mathrm{BPA}_{\delta}^{\mathrm{c}}$ has two sorts:

- the sort $\mathbf{P}$ of *processes*;

[1] For a definition of free κ-complete Boolean algebras, see e.g. [11].

[2] For a definition of regular cardinals, see e.g. [12,13]. They include $\aleph_0, \aleph_1, \aleph_2, \ldots$.

- the sort $\mathbf{C}$ of (*finite*) *conditions*.

The algebraic theory $\mathrm{BPA}_\delta^\mathrm{c}$ has the following constants and operators to build terms of sort $\mathbf{C}$:

- the *bottom* constant $\bot : \mathbf{C}$;
- the *top* constant $\top : \mathbf{C}$;
- for each $\eta \in \mathsf{C}_\mathsf{at}$, the *atomic condition* constant $\eta : \mathbf{C}$;
- the unary *complement* operator $- : \mathbf{C} \to \mathbf{C}$;
- the binary *join* operator $\sqcup : \mathbf{C} \times \mathbf{C} \to \mathbf{C}$;
- the binary *meet* operator $\sqcap : \mathbf{C} \times \mathbf{C} \to \mathbf{C}$.

The algebraic theory $\mathrm{BPA}_\delta^\mathrm{c}$ has the following constants and operators to build terms of sort $\mathbf{P}$:

- the *deadlock* constant $\delta : \mathbf{P}$;
- for each $a \in \mathsf{A}$, the *action* constant $a : \mathbf{P}$;
- the binary *alternative composition* operator $+ : \mathbf{P} \times \mathbf{P} \to \mathbf{P}$;
- the binary *sequential composition* operator $\cdot : \mathbf{P} \times \mathbf{P} \to \mathbf{P}$;
- the binary *guarded command* operator $:\to\, : \mathbf{C} \times \mathbf{P} \to \mathbf{P}$.

We use infix notation for the binary operators. The following precedence conventions are used to reduce the need for parentheses. The operators to build terms of sort $\mathbf{C}$ bind stronger than the operators to build terms of sort $\mathbf{P}$. The operator $\cdot$ binds stronger than all other binary operators to build terms of sort $\mathbf{P}$ and the operator $+$ binds weaker than all other binary operators to build terms of sort $\mathbf{P}$.

The constants and operators of $\mathrm{BPA}_\delta^\mathrm{c}$ to build terms of sort $\mathbf{P}$ are the constants and operators of BPA_δ and additionally the guarded command operator. Let p and q be closed terms of sort $\mathbf{P}$ and ζ be a closed term of sort $\mathbf{C}$. Intuitively, the constants and operators to build terms of sort $\mathbf{P}$ can be explained as follows:

- δ can neither perform an action nor terminate successfully;
- a first performs action a and then terminates successfully, both unconditionally;
- $p + q$ behaves either as p or as q, but not both;
- $p \cdot q$ first behaves as p, but when p terminates successfully it continues by behaving as q;
- $\zeta :\to p$ behaves as p under condition ζ.

Some earlier extensions of ACP include conditional expressions of the form $p \lhd \zeta \rhd q$; see e.g. [4]. This notation with triangles originates from [14]. We treat conditional expressions of the form $p \lhd \zeta \rhd q$, where p and q are terms of sort $\mathbf{P}$ and ζ is a term of sort $\mathbf{C}$, as abbreviations. That is, we write $p \lhd \zeta \rhd q$

Table 1
Axioms of Boolean algebras

$\phi \sqcup \bot = \phi$	BA1	$\phi \sqcap \top = \phi$	BA5
$\phi \sqcup -\phi = \top$	BA2	$\phi \sqcap -\phi = \bot$	BA6
$\phi \sqcup \psi = \psi \sqcup \phi$	BA3	$\phi \sqcap \psi = \psi \sqcap \phi$	BA7
$\phi \sqcup (\psi \sqcap \chi) = (\phi \sqcup \psi) \sqcap (\phi \sqcup \chi)$	BA4	$\phi \sqcap (\psi \sqcup \chi) = (\phi \sqcap \psi) \sqcup (\phi \sqcap \chi)$	BA8

Table 2
Axioms of $\mathrm{BPA}_\delta^\mathrm{c}$

$x + y = y + x$	A1	$\top :\to x = x$	GC1
$(x + y) + z = x + (y + z)$	A2	$\bot :\to x = \delta$	GC2
$x + x = x$	A3	$\phi :\to \delta = \delta$	GC3
$(x + y) \cdot z = x \cdot z + y \cdot z$	A4	$\phi :\to (x + y) = \phi :\to x + \phi :\to y$	GC4
$(x \cdot y) \cdot z = x \cdot (y \cdot z)$	A5	$\phi :\to x \cdot y = (\phi :\to x) \cdot y$	GC5
$x + \delta = x$	A6	$\phi :\to (\psi :\to x) = (\phi \sqcap \psi) :\to x$	GC6
$\delta \cdot x = \delta$	A7	$(\phi \sqcup \psi) :\to x = \phi :\to x + \psi :\to x$	GC7

for $\zeta :\to p + -\zeta :\to q$.

The axioms of $\mathrm{BPA}_\delta^\mathrm{c}$ are the axioms of Boolean Algebras (BA) given in Table 1 and the additional axioms given in Table 2. Axioms A1–A7 are the axioms of BPA_δ. So $\mathrm{BPA}_\delta^\mathrm{c}$ imports the axioms of both BA and BPA_δ. The axioms of BA given in Table 1 have been taken from [15]. Several alternatives for this axiomatization can be found in the literature (e.g. in [11,16]). If we use basic laws of BA other than axioms BA1–BA8, such as $\phi \sqcap \phi = \phi$ and $-(\phi \sqcap \psi) = -\phi \sqcup -\psi$, in a step of a derivation, we will refer to them as applications of BA and not give their derivation from axioms BA1–BA8. Axioms GC1–GC7 have been taken from [4], but with the axiom $x \cdot z \lhd \phi \rhd y \cdot z = (x \lhd \phi \rhd y) \cdot z$ (CO5) replaced by $\phi :\to x \cdot y = (\phi :\to x) \cdot y$ (GC5).

Example 1 *Consider a careful pedestrian who uses a crossing with traffic lights to cross a road with busy traffic safely. When the pedestrian arrives at the crossing and the light for pedestrians is green, he or she simply crosses the street. However, when the pedestrian arrives at the crossing and the light for pedestrians is red, he or she first makes a request for green light (e.g. by pushing a button) and then crosses the street when the light has changed. This behaviour can be described in* $\mathrm{BPA}_\delta^\mathrm{c}$ *as follows:*

$$PED = arrive \cdot (green :\to cross + red :\to (make\_req \cdot (green :\to cross)))\ .$$

The careful pedestrian described above does not cross the street if the light for

Table 3
Transition rules for $\mathrm{BPA}^{\mathrm{c}}_{\delta}$

$$\frac{}{a \xrightarrow{[\top]\,a} \surd}$$

$$\frac{x \xrightarrow{[\phi]\,a} \surd}{x + y \xrightarrow{[\phi]\,a} \surd} \qquad \frac{y \xrightarrow{[\phi]\,a} \surd}{x + y \xrightarrow{[\phi]\,a} \surd} \qquad \frac{x \xrightarrow{[\phi]\,a} x'}{x + y \xrightarrow{[\phi]\,a} x'} \qquad \frac{y \xrightarrow{[\phi]\,a} y'}{x + y \xrightarrow{[\phi]\,a} y'}$$

$$\frac{x \xrightarrow{[\phi]\,a} \surd}{x \cdot y \xrightarrow{[\phi]\,a} y} \qquad \frac{x \xrightarrow{[\phi]\,a} x'}{x \cdot y \xrightarrow{[\phi]\,a} x' \cdot y}$$

$$\frac{x \xrightarrow{[\phi]\,a} \surd}{\psi :\to x \xrightarrow{[\phi \sqcap \psi]\,a} \surd}\ \phi \sqcap \psi \neq \bot \qquad \frac{x \xrightarrow{[\phi]\,a} x'}{\psi :\to x \xrightarrow{[\phi \sqcap \psi]\,a} x'}\ \phi \sqcap \psi \neq \bot$$

pedestrians does not change from red to green after a request for green light. Whether the change from red to green will ever happen is not described here.

The terms of sort $\mathbf{C}$ are interpreted in $\mathcal{C}$ as usual.

We proceed to the presentation of the structural operational semantics of $\mathrm{BPA}^{\mathrm{c}}_{\delta}$. The following relations on closed terms of sort $\mathbf{P}$ are used:

- for each $\ell \in (\mathcal{C} \setminus \{\bot\}) \times \mathsf{A}$, a binary relation $\xrightarrow{\ell}$;
- for each $\ell \in (\mathcal{C} \setminus \{\bot\}) \times \mathsf{A}$, a unary relation $\xrightarrow{\ell} \surd$.

We write $p \xrightarrow{[\alpha]\,a} q$ instead of $(p, q) \in \xrightarrow{(\alpha,a)}$ and $p \xrightarrow{[\alpha]\,a} \surd$ instead of $p \in \xrightarrow{(\alpha,a)} \surd$. The relations $\xrightarrow{\ell} \surd$ and $\xrightarrow{\ell}$ can be explained as follows:

- $p \xrightarrow{[\alpha]\,a} \surd$: p is capable of performing action a under condition α and then terminating successfully;
- $p \xrightarrow{[\alpha]\,a} q$: p is capable of performing action a under condition α and then proceeding as q.

The structural operational semantics of $\mathrm{BPA}^{\mathrm{c}}_{\delta}$ is described by the transition rules given in Table 3. We will return to this structural operational semantics in Section 5.

3 Transition Systems and Splitting Bisimilarity for $\mathrm{BPA}^{\mathrm{c}}_{\delta}$

In this section, we introduce conditional transition systems and splitting bisimilarity of conditional transition systems. In Section 4, we will make use of conditional transition systems and splitting bisimilarity of conditional transition systems to construct models of $\mathrm{BPA}_\delta^\mathrm{c}$. In Section 5, we will show that the structural operational semantics presented in Section 2 induces a conditional transition system for each closed term of sort $\mathbf{P}$.

Conditional transition systems are labelled transition systems of which the labels consist of a condition different from $\bot$ and an action. Labels of this kind are sometimes called *guarded actions*. Henceforth, we write $\mathcal{C}_\kappa^-$ for $\mathcal{C}_\kappa \setminus \{\bot\}$.

Let κ be an infinite cardinal. Then a *κ-conditional transition system* T consists of the following:

- a set S of *states*;
- a set $\xrightarrow{\ell} \subseteq S \times S$, for each $\ell \in \mathcal{C}_\kappa^- \times \mathsf{A}$;
- a set $\xrightarrow{\ell} \surd \subseteq S$, for each $\ell \in \mathcal{C}_\kappa^- \times \mathsf{A}$;
- an *initial state* $s^0 \in S$.

If $(s, s') \in \xrightarrow{\ell}$ for some $\ell \in \mathcal{C}_\kappa^- \times \mathsf{A}$, then we say that there is a *transition* from s to s'. We usually write $s \xrightarrow{[\alpha]\,a} s'$ instead of $(s, s') \in \xrightarrow{(\alpha,a)}$ and $s \xrightarrow{[\alpha]\,a} \surd$ instead of $s \in \xrightarrow{(\alpha,a)} \surd$. Furthermore, we write $\rightarrow$ for the family of sets $(\xrightarrow{\ell})_{\ell \in \mathcal{C}_\kappa^- \times \mathsf{A}}$ and $\rightarrow\surd$ for the family of sets $(\xrightarrow{\ell}\surd)_{\ell \in \mathcal{C}_\kappa^- \times \mathsf{A}}$.

The relations $\xrightarrow{\ell}\surd$ and $\xrightarrow{\ell}$ can be explained as follows:

- $s \xrightarrow{[\alpha]\,a} \surd$: in state s, it is possible to perform action a under condition α, and by doing so to terminate successfully;
- $s \xrightarrow{[\alpha]\,a} s'$: in state s, it is possible to perform action a under condition α, and by doing so to make a transition to state s'.

A conditional transition system may have states that are not reachable from its initial state by a number of transitions. Unreachable states, and the transitions between them, are not relevant to the behaviour represented by the transition system. Connected conditional transition systems are transition systems without unreachable states.

Let $T = (S, \rightarrow, \rightarrow\surd, s^0)$ be a κ-conditional transition system (for an infinite cardinal κ). Then the *reachability* relation of T is the smallest relation $\twoheadrightarrow \subseteq S \times S$ such that:

- $s \twoheadrightarrow s$;
- if $s \xrightarrow{\ell} s'$ and $s' \twoheadrightarrow s''$, then $s \twoheadrightarrow s''$.

We write $\mathrm{RS}(T)$ for $\{s \in S \mid s^0 \twoheadrightarrow s\}$. T is called a *connected* κ-conditional transition system if $S = \mathrm{RS}(T)$. Henceforth, we will only consider connected conditional transition systems. However, this often calls for extraction of the connected part of a conditional transition system that is composed of connected conditional transition systems.

Let $T = (S, \rightarrow, \rightarrow\surd, s^0)$ be a κ-conditional transition system (for an infinite cardinal κ) that is not necessarily connected. Then the *connected part* of T, written $\Gamma(T)$, is defined as follows:

$$\Gamma(T) = (S', \rightarrow', \rightarrow\surd', s^0) ,$$

where

$$S' = \mathrm{RS}(T) ,$$

and for every $\ell \in \mathcal{C}_\kappa^- \times \mathsf{A}$:

$$\begin{aligned} \xrightarrow{\ell}{}' &= \xrightarrow{\ell} \cap (S' \times S') , \\ \xrightarrow{\ell}\surd' &= \xrightarrow{\ell}\surd \cap S' . \end{aligned}$$

It is assumed that for each infinite cardinal κ a fixed but arbitrary set $\mathcal{S}_\kappa$ with the following properties has been given:

- the cardinality of $\mathcal{S}_\kappa$ is greater than or equal to κ;
- if $S_1, S_2 \subseteq \mathcal{S}_\kappa$, then $S_1 \uplus S_2 \subseteq \mathcal{S}_\kappa$ and $S_1 \times S_2 \subseteq \mathcal{S}_\kappa$.[3]

Let κ be an infinite cardinal. Then $\mathbb{CTS}_\kappa$ is the set of all connected κ-conditional transition systems $T = (S, \rightarrow, \rightarrow\surd, s^0)$ such that $S \subset \mathcal{S}_\kappa$ and the branching degree of T is less than κ, i.e. for all $s \in S$, the cardinality of the set $\{(\ell, s') \in (\mathcal{C}_\kappa^- \times \mathsf{A}) \times S \mid (s, s') \in \xrightarrow{\ell}\} \cup \{\ell \in \mathcal{C}_\kappa^- \times \mathsf{A} \mid s \in \xrightarrow{\ell}\surd\}$ is less than κ.

The condition $S \subset \mathcal{S}_\kappa$ guarantees that $\mathbb{CTS}_\kappa$ is indeed a set.

A conditional transition system is said to be *finitely branching* if its branching degree is less than $\aleph_0$. Otherwise, it is said to be *infinitely branching*.

The identity of the states of a conditional transition system is not relevant to the behaviour represented by it. Conditional transition systems that differ only with respect to the identity of the states are isomorphic.

[3] We write $A \uplus B$ for the disjoint union of sets A and B, i.e. $A \uplus B = (A \times \{\emptyset\}) \cup (B \times \{\{\emptyset\}\})$. We write μ_1 and μ_2 for the associated injections $\mu_1 : A \rightarrow A \uplus B$ and $\mu_2 : B \rightarrow A \uplus B$, defined by $\mu_1(a) = (a, \emptyset)$ and $\mu_2(b) = (b, \{\emptyset\})$.

Let $T_1 = (S_1, \rightarrow_1, \rightarrow\surd_1, s_1^0)$ and $T_2 = (S_2, \rightarrow_2, \rightarrow\surd_2, s_2^0)$ be κ-conditional transition systems (for an infinite cardinal κ). Then T_1 and T_2 are *isomorphic*, written $T_1 \cong T_2$, if there exists a bijective function $b : S_1 \rightarrow S_2$ such that:

- $b(s_1^0) = s_2^0$;
- $s_1 \xrightarrow{\ell}_1 s_1'$ iff $b(s_1) \xrightarrow{\ell}_2 b(s_1')$;
- $s \xrightarrow{\ell} \surd_1$ iff $b(s) \xrightarrow{\ell} \surd_2$.

Henceforth, we will always consider two conditional transition systems essentially the same if they are isomorphic.

Remark 2 *The set $\mathbb{CTS}_\kappa$ is independent of $\mathcal{S}_\kappa$. By that we mean the following. Let $\mathbb{CTS}_\kappa$ and $\mathbb{CTS}_\kappa'$ result from different choices for $\mathcal{S}_\kappa$. Then there exists a bijection $b : \mathbb{CTS}_\kappa \rightarrow \mathbb{CTS}_\kappa'$ such that for all $T \in \mathbb{CTS}_\kappa$, $T \cong b(T)$.*

Bisimilarity has to be adapted to the setting with guarded actions. In the definition given below, we use two well-known notions from the field of Boolean algebras: a partial order relation $\sqsubseteq$ on $\mathcal{C}_\kappa$ and a unary operation $\bigsqcup$ on the set of all subsets of $\mathcal{C}_\kappa$ of cardinality less than κ (for each infinite cardinal κ). The relation $\sqsubseteq$ and the operation $\bigsqcup$ are defined by

$$\alpha \sqsubseteq \beta \text{ iff } \alpha \sqcup \beta = \beta \qquad \text{and} \qquad \textstyle\bigsqcup C \text{ is the supremum of } C \text{ in } (\mathcal{C}_\kappa, \sqsubseteq)\ ,$$

respectively. The operation $\bigsqcup$ is defined for all subsets of $\mathcal{C}_\kappa$ of cardinality less than κ because $\mathcal{C}_\kappa$ is κ-complete.

Let $T_1 = (S_1, \rightarrow_1, \rightarrow\surd_1, s_1^0) \in \mathbb{CTS}_\kappa$ and $T_2 = (S_2, \rightarrow_2, \rightarrow\surd_2, s_2^0) \in \mathbb{CTS}_\kappa$ (for an infinite cardinal κ). Then a *splitting bisimulation* B between T_1 and T_2 is a binary relation $B \subseteq S_1 \times S_2$ such that $B(s_1^0, s_2^0)$ and for all s_1, s_2 such that $B(s_1, s_2)$:

- if $s_1 \xrightarrow{[\alpha]\,a}_1 s_1'$, then there is a set $CS_2' \subseteq \mathcal{C}_\kappa^- \times S_2$ of cardinality less than κ such that $\alpha \sqsubseteq \bigsqcup \mathrm{dom}(CS_2')$ and for all $(\alpha', s_2') \in CS_2'$, $s_2 \xrightarrow{[\alpha']\,a}_2 s_2'$ and $B(s_1', s_2')$;
- if $s_2 \xrightarrow{[\alpha]\,a}_2 s_2'$, then there is a set $CS_1' \subseteq \mathcal{C}_\kappa^- \times S_1$ of cardinality less than κ such that $\alpha \sqsubseteq \bigsqcup \mathrm{dom}(CS_1')$ and for all $(\alpha', s_1') \in CS_1'$, $s_1 \xrightarrow{[\alpha']\,a}_1 s_1'$ and $B(s_1', s_2')$;
- if $s_1 \xrightarrow{[\alpha]\,a} \surd_1$, then there is a set $C' \subseteq \mathcal{C}_\kappa^-$ of cardinality less than κ such that $\alpha \sqsubseteq \bigsqcup C'$ and for all $\alpha' \in C'$, $s_2 \xrightarrow{[\alpha']\,a} \surd_2$;
- if $s_2 \xrightarrow{[\alpha]\,a} \surd_2$, then there is a set $C' \subseteq \mathcal{C}_\kappa^-$ of cardinality less than κ such that $\alpha \sqsubseteq \bigsqcup C'$ and for all $\alpha' \in C'$, $s_1 \xrightarrow{[\alpha']\,a} \surd_1$.

Two conditional transition systems $T_1, T_2 \in \mathbb{CTS}_\kappa$ are *splitting bisimilar*, written $T_1 \mathrel{\underline{\leftrightarrow}} T_2$, if there exists a splitting bisimulation B between T_1 and T_2. Let B be a splitting bisimulation between T_1 and T_2. Then we say that B is a splitting bisimulation *witnessing* $T_1 \mathrel{\underline{\leftrightarrow}} T_2$.

The name splitting bisimulation is used because a transition of one of the related transition systems may be simulated by a set of transitions of the other transition system. Splitting bisimulation should not be confused with split bisimulation [17]. We think that splitting bisimulation can be reformulated in a style that is similar to the style in which probabilistic bisimulation is formulated in [18]. We refrain from such a reformulation because it would require the introduction of various auxiliary notions and notations.

It is easy to see that $\mathrel{\underline{\leftrightarrow}}$ is an equivalence on $\mathbb{CTS}_\kappa$. Let $T \in \mathbb{CTS}_\kappa$. Then we write $[\,T\,]_{\mathrel{\underline{\leftrightarrow}}}$ for $\{T' \in \mathbb{CTS}_\kappa \mid T \mathrel{\underline{\leftrightarrow}} T'\}$, i.e. the $\mathrel{\underline{\leftrightarrow}}$-equivalence class of T. We write $\mathbb{CTS}_\kappa/{\mathrel{\underline{\leftrightarrow}}}$ for the set of equivalence classes $\{[\,T\,]_{\mathrel{\underline{\leftrightarrow}}} \mid T \in \mathbb{CTS}_\kappa\}$.

In Section 4, we will use $\mathbb{CTS}_\kappa/{\mathrel{\underline{\leftrightarrow}}}$ as domain of a structure that is a model of $\mathrm{BPA}_\delta^\mathrm{c}$. As domain of a structure, $\mathbb{CTS}_\kappa/{\mathrel{\underline{\leftrightarrow}}}$ must be a set. That is the case because $\mathbb{CTS}_\kappa$ is a set. The latter is guaranteed by considering only conditional transition systems of which the set of states is a subset of $\mathcal{S}_\kappa$.

Remark 3 *The question arises whether $\mathcal{S}_\kappa$ is large enough if its cardinality is greater than or equal to κ. This question can be answered in the affirmative. Let $T = (S, \rightarrow, \rightarrow\surd, s^0)$ be a connected κ-conditional transition system of which the branching degree is less than κ. Then there exists a connected κ-conditional transition system $T' = (S', \rightarrow', \rightarrow\surd', s^{0\prime})$ of which the branching degree is less than κ such that $T \mathrel{\underline{\leftrightarrow}} T'$ and the cardinality of S' is less than κ.*

It is easy to see that, if we would consider conditional transition systems with unreachable states as well, each conditional transition system would be splitting bisimilar to its connected part. This justifies the choice to consider only connected conditional transition systems. It is easy to see that isomorphic conditional transition systems are splitting bisimilar. This justifies the choice to consider conditional transition systems essentially the same if they are isomorphic.

In the remainder of this section, we sketch how splitting bisimilarity is related to ordinary bisimilarity.

Let $T = (S, \rightarrow, \rightarrow\surd, s^0) \in \mathbb{CTS}_\kappa$ (for an infinite cardinal κ). We write $\simeq_T$ for the maximal splitting bisimulation witnessing $T \mathrel{\underline{\leftrightarrow}} T$ (such a relation always exists). It is easy to see that $\simeq_T$ is an equivalence relation on S. It identifies states of T that can simulate the conditional transitions of each other. The *condition-normal form* of T, written $\mathrm{CN}(T)$, is defined as follows:

$$\mathrm{CN}(T) = (S, \rightarrow', \rightarrow\surd', s^0)\ ,$$

where for every $(\alpha, a) \in \mathcal{C}_\kappa^- \times \mathsf{A}$:

$$\begin{aligned}
\xrightarrow{(\alpha,a)}{}' &= \Big\{(s, s') \;\Big|\; \exists \beta \bullet s \xrightarrow{[\beta]\,a} s' \land \\
&\qquad\qquad \alpha = \bigsqcup\Big\{\beta' \;\Big|\; \exists s'' \bullet \Big(s' \simeq_T s'' \land s \xrightarrow{[\beta']\,a} s''\Big)\Big\}\Big\} , \\
\xrightarrow{(\alpha,a)} \surd' &= \Big\{s \;\Big|\; \exists \beta \bullet s \xrightarrow{[\beta]\,a} \surd \land \alpha = \bigsqcup\Big\{\beta' \;\Big|\; s \xrightarrow{[\beta']\,a} \surd\Big\}\Big\} .
\end{aligned}$$

It is easy to see that $\mathrm{CN}(T) \in \mathbb{CTS}_\kappa$ and $T \Leftrightarrow \mathrm{CN}(T)$. We have $T = \mathrm{CN}(T)$ iff T has the following properties:

- if $s_1 \xrightarrow{[\alpha]\,a} s_1'$, $s_1 \xrightarrow{[\beta]\,a} s_1''$ and $s_1' \simeq_T s_1''$, then $\alpha = \beta$;
- if $s_1 \xrightarrow{[\alpha]\,a} \surd$ and $s_1 \xrightarrow{[\beta]\,a} \surd$, then $\alpha = \beta$.

We say that T is *condition-normal* if $T = \mathrm{CN}(T)$.

Let $T_1 = (S_1, \rightarrow_1, \rightarrow\surd_1, s_1^0) \in \mathbb{CTS}_\kappa$ and $T_2 = (S_2, \rightarrow_2, \rightarrow\surd_2, s_2^0) \in \mathbb{CTS}_\kappa$ (for an infinite cardinal κ). Then a *bisimulation* B between T_1 and T_2 is a binary relation $B \subseteq S_1 \times S_2$ such that $B(s_1^0, s_2^0)$ and for all s_1, s_2 such that $B(s_1, s_2)$:

- if $s_1 \xrightarrow{\ell}_1 s_1'$, then there is a $s_2' \in S_2$ such that $s_2 \xrightarrow{\ell}_2 s_2'$ and $B(s_1', s_2')$;
- if $s_2 \xrightarrow{\ell}_2 s_2'$, then there is a $s_1' \in S_1$ such that $s_1 \xrightarrow{\ell}_1 s_1'$ and $B(s_1', s_2')$;
- $s_1 \xrightarrow{\ell} \surd_1$ iff $s_2 \xrightarrow{\ell} \surd_2$.

Two conditional transition systems $T_1, T_2 \in \mathbb{CTS}_\kappa$ are *bisimilar*, written $T_1 \leftrightarrow T_2$, if there exists a bisimulation B between T_1 and T_2. We have $\mathrm{CN}(T_1) \Leftrightarrow \mathrm{CN}(T_2)$ iff $\mathrm{CN}(T_1) \leftrightarrow \mathrm{CN}(T_2)$. So, splitting bisimilarity and ordinary bisimilarity coincide on condition-normal conditional transition systems. It is worth mentioning that we do not have this result if we replace $s' \simeq_T s''$ by $s' = s''$ in the definition of CN.

4 Full Splitting Bisimulation Models of BPA$_\delta^{\mathrm{c}}$

In this section, we introduce the full splitting bisimulation models of BPA$_\delta^{\mathrm{c}}$. They are models in which equivalence classes of conditional transition systems modulo splitting bisimilarity are taken as processes. The qualification "full" originates from [19]. It expresses that there exist other splitting bisimulation models, but each of them is isomorphically embedded in a full splitting bisimulation model.

There is a full splitting bisimulation model of BPA$_\delta^{\mathrm{c}}$ for each infinite cardinal. To obtain the full splitting bisimulation model of BPA$_\delta^{\mathrm{c}}$ for a fixed infinite cardinal κ, we associate the set $\mathbb{CTS}_\kappa/{\Leftrightarrow}$ with the sort $\mathbf{P}$, an element of $\mathbb{CTS}_\kappa/{\Leftrightarrow}$

with each of the constants δ and a ($a \in \mathsf{A}$), and an operation on $\mathbb{CTS}_\kappa/\Leftrightarrow$ with each of the operators $+$, $\cdot$ and $:\rightarrow$.[4] We begin by associating elements of $\mathbb{CTS}_\kappa$ and operations on $\mathbb{CTS}_\kappa$ with these constants and operators. The result of this is subsequently lifted to $\mathbb{CTS}_\kappa/\Leftrightarrow$.

It is assumed that for each infinite cardinal κ a fixed but arbitrary choice function $\mathrm{ch}_\kappa : (\mathcal{P}(\mathcal{S}_\kappa) \setminus \emptyset) \to \mathcal{S}_\kappa$ such that for all $S \in \mathcal{P}(\mathcal{S}_\kappa) \setminus \emptyset$, $\mathrm{ch}_\kappa(S) \in S$ has been given. The function ch_κ is used whenever there is a need to get a fresh state from $\mathcal{S}_\kappa$.

We associate with each constant c mentioned above an element $\widehat{c}$ of $\mathbb{CTS}_\kappa$ and with each operator f mentioned above an operation $\widehat{f}$ on $\mathbb{CTS}_\kappa$ as follows.

- $$\widehat{\delta} = (\{s^0\}, \emptyset, \emptyset, s^0) \;,$$
  where
  $$s^0 = \mathrm{ch}_\kappa(\mathcal{S}_\kappa) \;.$$
- $$\widehat{a} = (\{s^0\}, \emptyset, \rightarrow\surd, s^0) \;,$$
  where
  $$\begin{aligned} s^0 &= \mathrm{ch}_\kappa(\mathcal{S}_\kappa) \;, \\ \xrightarrow{(\top,a)}\surd &= \{s^0\} \;, \end{aligned}$$
  and for every $(\alpha, a') \in (\mathcal{C}_\kappa^- \times \mathsf{A}) \setminus \{(\top, a)\}$:
  $$\xrightarrow{(\alpha,a')}\surd = \emptyset \;.$$
- Let $T_i = (S_i, \rightarrow_i, \rightarrow\surd_i, s_i^0) \in \mathbb{CTS}_\kappa$ for $i = 1, 2$. Then
  $$T_1 \mathbin{\widehat{+}} T_2 = \Gamma(S, \rightarrow, \rightarrow\surd, s^0) \;,$$
  where
  $$\begin{aligned} s^0 &= \mathrm{ch}_\kappa(\mathcal{S}_\kappa \setminus (S_1 \uplus S_2)) \;, \\ S &= \{s^0\} \cup (S_1 \uplus S_2) \;, \end{aligned}$$

[4] In this paper, we loosely include the operation associated with the operator $:\rightarrow$ in the operations on $\mathbb{CTS}_\kappa/\Leftrightarrow$. Actually, it is an operation from $\mathcal{C} \times \mathbb{CTS}_\kappa/\Leftrightarrow$ to $\mathbb{CTS}_\kappa/\Leftrightarrow$.

and for every $(\alpha, a) \in \mathcal{C}_\kappa^- \times \mathsf{A}$:

$$
\begin{aligned}
\xrightarrow{(\alpha,a)} \;&= \left\{(s^0, \mu_1(s)) \;\middle|\; s_1^0 \xrightarrow{[\alpha]\,a}_1 s\right\} \\
&\cup \left\{(s^0, \mu_2(s)) \;\middle|\; s_2^0 \xrightarrow{[\alpha]\,a}_2 s\right\} \\
&\cup \left\{(\mu_1(s), \mu_1(s')) \;\middle|\; s \xrightarrow{[\alpha]\,a}_1 s'\right\} \\
&\cup \left\{(\mu_2(s), \mu_2(s')) \;\middle|\; s \xrightarrow{[\alpha]\,a}_2 s'\right\} , \\
\xrightarrow{(\alpha,a)}\surd &= \left\{s^0 \;\middle|\; s_1^0 \xrightarrow{[\alpha]\,a} \surd_1\right\} \\
&\cup \left\{s^0 \;\middle|\; s_2^0 \xrightarrow{[\alpha]\,a} \surd_2\right\} \\
&\cup \left\{\mu_1(s) \;\middle|\; s \xrightarrow{[\alpha]\,a} \surd_1\right\} \\
&\cup \left\{\mu_2(s) \;\middle|\; s \xrightarrow{[\alpha]\,a} \surd_2\right\} .
\end{aligned}
$$

- Let $T_i = (S_i, \rightarrow_i, \rightarrow\surd_i, s_i^0) \in \mathbb{CTS}_\kappa$ for $i = 1, 2$. Then

  $$T_1 \mathbin{\hat{\cdot}} T_2 = \Gamma(S, \rightarrow, \rightarrow\surd, s^0) ,$$

  where

  $$
  \begin{aligned}
  S \;&= S_1 \uplus S_2 , \\
  s^0 &= \mu_1(s_1^0) ,
  \end{aligned}
  $$

  and for every $(\alpha, a) \in \mathcal{C}_\kappa^- \times \mathsf{A}$:

  $$
  \begin{aligned}
  \xrightarrow{(\alpha,a)} \;&= \left\{(\mu_1(s), \mu_1(s')) \;\middle|\; s \xrightarrow{[\alpha]\,a}_1 s'\right\} \\
  &\cup \left\{(\mu_1(s), \mu_2(s_2^0)) \;\middle|\; s \xrightarrow{[\alpha]\,a} \surd_1\right\} \\
  &\cup \left\{(\mu_2(s), \mu_2(s')) \;\middle|\; s \xrightarrow{[\alpha]\,a}_2 s'\right\} , \\
  \xrightarrow{(\alpha,a)}\surd &= \left\{\mu_2(s) \;\middle|\; s \xrightarrow{[\alpha]\,a} \surd_2\right\} .
  \end{aligned}
  $$

- Let $\alpha \in \mathcal{C}$ and $T = (S, \rightarrow, \rightarrow\surd, s^0) \in \mathbb{CTS}_\kappa$. Then

  $$\alpha \mathbin{\widehat{:\rightarrow}} T = \Gamma(S, \rightarrow', \rightarrow\surd', s^0) ,$$

  where for every $(\alpha', a) \in \mathcal{C}_\kappa^- \times \mathsf{A}$:

  $$
  \begin{aligned}
  \xrightarrow{(\alpha',a)}{}' \;&= \left\{(s^0, s') \;\middle|\; \exists\beta \bullet \left(s^0 \xrightarrow{[\beta]\,a} s' \land \alpha' = \alpha \sqcap \beta\right)\right\} \\
  &\cup \left\{(s, s') \;\middle|\; s \xrightarrow{[\alpha']\,a} s' \land s \neq s^0\right\} , \\
  \xrightarrow{(\alpha',a)}\surd' &= \left\{s^0 \;\middle|\; \exists\beta \bullet \left(s^0 \xrightarrow{[\beta]\,a} \surd \land \alpha' = \alpha \sqcap \beta\right)\right\} \\
  &\cup \left\{s \;\middle|\; s \xrightarrow{[\alpha']\,a} \surd \land s \neq s^0\right\} .
  \end{aligned}
  $$

In the definition of alternative composition on $\mathbb{CTS}_\kappa$, the connected part of a conditional transition system is extracted because the initial states of the conditional transition systems T_1 and T_2 may be unreachable from the new initial state. The new initial state is introduced because, in T_1 and/or T_2, there may exist a transition back to the initial state. In the definition of sequential composition on $\mathbb{CTS}_\kappa$, the connected part of a conditional transition system is extracted because the initial state of the conditional transition system T_2 may be unreachable from the initial state of the conditional transition system T_1 due to absence of termination in T_1.

Remark 4 *The elements of $\mathbb{CTS}_\kappa$ and the operations on $\mathbb{CTS}_\kappa$ defined above are independent of ch_κ. Different choices for ch_κ lead for each constant to isomorphic elements of $\mathbb{CTS}_\kappa$ and lead for each operator to operations on $\mathbb{CTS}_\kappa$ with isomorphic results.*

We can easily show that splitting bisimilarity is a congruence with respect to alternative composition, sequential composition and guarded command.

Proposition 5 (Congruence) *Let κ be an infinite cardinal. Then for all $T_1, T_2, T'_1, T'_2 \in \mathbb{CTS}_\kappa$ and $\alpha \in \mathcal{C}$, $T_1 \mathrel{\underline{\Leftrightarrow}} T'_1$ and $T_2 \mathrel{\underline{\Leftrightarrow}} T'_2$ imply $T_1 \mathbin{\hat{+}} T_2 \mathrel{\underline{\Leftrightarrow}} T'_1 \mathbin{\hat{+}} T'_2$, $T_1 \mathbin{\hat{\cdot}} T_2 \mathrel{\underline{\Leftrightarrow}} T'_1 \mathbin{\hat{\cdot}} T'_2$ and $\alpha \mathbin{\widehat{:\rightarrow}} T_1 \mathrel{\underline{\Leftrightarrow}} \alpha \mathbin{\widehat{:\rightarrow}} T'_1$.*

PROOF. Let $T_i = (S_i, \rightarrow_i, \rightarrow\surd_i, s_i^0)$ and $T'_i = (S'_i, \rightarrow'_i, \rightarrow\surd'_i, s_i^{0\prime})$ for $i = 1, 2$. Let R_1 and R_2 be splitting bisimulations witnessing $T_1 \mathrel{\underline{\Leftrightarrow}} T'_1$ and $T_2 \mathrel{\underline{\Leftrightarrow}} T'_2$, respectively. Then we construct relations $R_{\hat{+}}$, $R_{\hat{\cdot}}$ and $R_{\widehat{:\rightarrow}}$ as follows:

- $R_{\hat{+}} = (\{(s^0, s^{0\prime})\} \cup \mu_1(R_1) \cup \mu_2(R_2)) \cap (S \times S')$, where S and S' are the sets of states of $T_1 \mathbin{\hat{+}} T_2$ and $T'_1 \mathbin{\hat{+}} T'_2$, respectively, and s^0 and $s^{0\prime}$ are the initial states of $T_1 \mathbin{\hat{+}} T_2$ and $T'_1 \mathbin{\hat{+}} T'_2$, respectively;
- $R_{\hat{\cdot}} = (\mu_1(R_1) \cup \mu_2(R_2)) \cap (S \times S')$, where S and S' are the sets of states of $T_1 \mathbin{\hat{\cdot}} T_2$ and $T'_1 \mathbin{\hat{\cdot}} T'_2$, respectively;
- $R_{\widehat{:\rightarrow}} = R_1 \cap (S \times S')$, where S and S' are the sets of states of $\alpha \mathbin{\widehat{:\rightarrow}} T_1$ and $\alpha \mathbin{\widehat{:\rightarrow}} T'_1$, respectively.

Here, we write $\mu_i(R_i)$ for $\{(\mu_i(s), \mu_i(s')) \mid R_i(s, s')\}$, where μ_i is used to denote both the injection of S_i into $S_1 \uplus S_2$ and the injection of S'_i into $S'_1 \uplus S'_2$. Given the definitions of alternative composition, sequential composition and guarded command, it is easy to see that $R_{\hat{+}}$, $R_{\hat{\cdot}}$ and $R_{\widehat{:\rightarrow}}$ are splitting bisimulations witnessing $T_1 \mathbin{\hat{+}} T_2 \mathrel{\underline{\Leftrightarrow}} T'_1 \mathbin{\hat{+}} T'_2$, $T_1 \mathbin{\hat{\cdot}} T_2 \mathrel{\underline{\Leftrightarrow}} T'_1 \mathbin{\hat{\cdot}} T'_2$ and $\alpha \mathbin{\widehat{:\rightarrow}} T_1 \mathrel{\underline{\Leftrightarrow}} \alpha \mathbin{\widehat{:\rightarrow}} T'_1$, respectively. $\Box$

The *full splitting bisimulation models* $\mathfrak{P}_\kappa^{\mathrm{c}}$ of $\mathrm{BPA}_\delta^{\mathrm{c}}$, one for each infinite cardinal

κ, are the expansions of $\mathcal{C}$ with:[5]

- for the sort $\mathbf{P}$, a non-empty set $\mathcal{P}$;[6]
- for the constant δ, an element $\widetilde{\delta}$ of $\mathcal{P}$;
- for each constant a ($a \in \mathsf{A}$), an element $\widetilde{a}$ of $\mathcal{P}$;
- for the operator $+$, an operation $\widetilde{+} : \mathcal{P} \times \mathcal{P} \to \mathcal{P}$;
- for the operator $\cdot$, an operation $\widetilde{\cdot} : \mathcal{P} \times \mathcal{P} \to \mathcal{P}$;
- for the operator $:\rightarrow$, an operation $\widetilde{:\rightarrow} : \mathcal{C} \times \mathcal{P} \to \mathcal{P}$;

where those ingredients are defined as follows:

$$
\begin{array}{lll}
\mathcal{P} = \mathbb{CTS}_\kappa / \underline{\Leftrightarrow}\,, & [\,T_1\,]_{\underline{\Leftrightarrow}} \,\widetilde{+}\, [\,T_2\,]_{\underline{\Leftrightarrow}} & = [\,T_1 \,\widehat{+}\, T_2\,]_{\underline{\Leftrightarrow}}\,, \\
\widetilde{\delta} \; = [\,\widehat{\delta}\,]_{\underline{\Leftrightarrow}}\,, & [\,T_1\,]_{\underline{\Leftrightarrow}} \,\widetilde{\cdot}\, [\,T_2\,]_{\underline{\Leftrightarrow}} & = [\,T_1 \,\widehat{\cdot}\, T_2\,]_{\underline{\Leftrightarrow}}\,, \\
\widetilde{a} \; = [\,\widehat{a}\,]_{\underline{\Leftrightarrow}}\,, & \alpha \,\widetilde{:\rightarrow}\, [\,T_1\,]_{\underline{\Leftrightarrow}} & = [\,\alpha \,\widehat{:\rightarrow}\, T_1\,]_{\underline{\Leftrightarrow}}\,.
\end{array}
$$

Alternative composition, sequential composition and guarded command on $\mathbb{CTS}_\kappa/\underline{\Leftrightarrow}$ are well-defined because $\underline{\Leftrightarrow}$ is a congruence with respect to the corresponding operations on $\mathbb{CTS}_\kappa$. $\mathbb{CTS}_\kappa/\underline{\Leftrightarrow}$ is called the *process domain* of $\mathfrak{P}^{\mathrm{c}}_\kappa$.

The structures $\mathfrak{P}^{\mathrm{c}}_\kappa$ are models of $\mathrm{BPA}^{\mathrm{c}}_\delta$.

Theorem 6 (Soundness of BPA$^{\mathrm{c}}_\delta$) *For each infinite cardinal κ, we have $\mathfrak{P}^{\mathrm{c}}_\kappa \models \mathrm{BPA}^{\mathrm{c}}_\delta$.*

PROOF. Because $\mathfrak{P}^{\mathrm{c}}_\kappa$ is an expansion of $\mathcal{C}$, it is not necessary to show that the axioms of BA are sound. The soundness of all remaining axioms follows easily from the definitions of the ingredients of $\mathfrak{P}^{\mathrm{c}}_\kappa$. $\square$

As to be expected, the full splitting bisimulation models are related by isomorphic embeddings.

Theorem 7 (Isomorphic Embedding) *Let κ and κ' be infinite cardinals such that $\kappa < \kappa'$. Then $\mathfrak{P}^{\mathrm{c}}_\kappa$ is isomorphically embedded in $\mathfrak{P}^{\mathrm{c}}_{\kappa'}$.*

PROOF. It follows immediately from the definitions of $\mathbb{CTS}_\kappa$, $\mathbb{CTS}_{\kappa'}$ and $\underline{\Leftrightarrow}$ that for each $P \in \mathbb{CTS}_\kappa/\underline{\Leftrightarrow}$, there exists a unique $P' \in \mathbb{CTS}_{\kappa'}/\underline{\Leftrightarrow}$ such that $P \subseteq P'$. Now consider the function $h : \mathbb{CTS}_\kappa/\underline{\Leftrightarrow} \to \mathbb{CTS}_{\kappa'}/\underline{\Leftrightarrow}$ where for each

[5] $\mathfrak{P}$ is the Gothic capital P.

[6] Here, the expansions involve the addition of a domain because they go from a one-sorted algebra to a two-sorted algebra.

$P \in \mathbb{CTS}_\kappa/{\Leftrightarrow}$, $h(P)$ is the unique $P' \in \mathbb{CTS}_{\kappa'}/{\Leftrightarrow}$ such that $P \subseteq P'$. It follows immediately from the definition of h that h is injective. Moreover, it follows easily from the definitions of the operations on $\mathbb{CTS}_\kappa/{\Leftrightarrow}$ and $\mathbb{CTS}_{\kappa'}/{\Leftrightarrow}$ that h, together with the identity function on $\mathcal{C}$, is a homomorphism from $\mathfrak{P}^{\mathrm{c}}_\kappa$ to $\mathfrak{P}^{\mathrm{c}}_{\kappa'}$. $\square$

5 SOS-Based Splitting Bisimilarity for $\mathbf{BPA}^{\mathrm{c}}_\delta$

It is customary to associate transition systems with closed terms (of sort **P**) from the language of an ACP-like theory about processes by means of structural operational semantics and to identify closed terms if their associated transition systems are equivalent by a bisimilarity-based notion of equivalence.

The structural operational semantics of $\mathrm{BPA}^{\mathrm{c}}_\delta$ presented in Section 2 determines a conditional transition system for each process that can be denoted by a closed term of sort **P**. These transition systems are special in the sense that their states are closed terms of sort **P**.

Let p be a closed term of sort **P**. Then the transition system of p *induced by* the structural operational semantics of $\mathrm{BPA}^{\mathrm{c}}_\delta$, written $\mathrm{CTS}(p)$, is the connected conditional transition system $\Gamma(S, \rightarrow, \rightarrow\surd, s^0)$, where:

- S is the set of all closed terms of sort **P**;
- $\xrightarrow{(\alpha,a)} \subseteq S \times S$ and $\xrightarrow{(\alpha,a)}\surd \subseteq S$ for each $\alpha \in \mathcal{C} \setminus \{\bot\}$ and $a \in \mathsf{A}$ are the smallest subsets of $S \times S$ and S, respectively, for which the transition rules from Table 3 hold;
- $s^0 \in S$ is the closed term p.

Let p and q be closed terms of sort **P**. Then we say that p and q are *splitting bisimilar*, written $p \Leftrightarrow q$, if $\mathrm{CTS}(p) \Leftrightarrow \mathrm{CTS}(q)$.

Clearly, the structural operational semantics does not give rise to infinitely branching conditional transition systems. For each closed term p of sort **P**, there exists a $T \in \mathbb{CTS}_{\aleph_0}$ such that $\mathrm{CTS}(p) \cong T$. In Section 4, it has been shown that it is possible to consider infinitely branching conditional transition systems too.

6 ACP with Conditions

In order to support parallelism and communication, we add parallel composition and encapsulation operators to $\mathrm{BPA}^{\mathrm{c}}_\delta$, resulting in $\mathrm{ACP}^{\mathrm{c}}$.

Like in $\mathrm{BPA}_\delta^{\mathrm{c}}$, it is assumed that a fixed but arbitrary finite set of *actions* A, with $\delta \notin \mathsf{A}$, and a fixed but arbitrary set of *atomic conditions* C_{at} have been given. We write A_δ for $\mathsf{A} \cup \{\delta\}$. In $\mathrm{ACP}^{\mathrm{c}}$, it is further assumed that a fixed but arbitrary commutative and associative *communication* function $| : \mathsf{A}_\delta \times \mathsf{A}_\delta \to \mathsf{A}_\delta$, such that $\delta \,|\, a = \delta$ for all $a \in \mathsf{A}_\delta$, has been given. The function $|$ is regarded to give the result of synchronously performing any two actions for which this is possible, and to be δ otherwise.

The theory $\mathrm{ACP}^{\mathrm{c}}$ is an extension of $\mathrm{BPA}_\delta^{\mathrm{c}}$. It has the constants and operators of $\mathrm{BPA}_\delta^{\mathrm{c}}$ and in addition:

- the binary *parallel composition* operator $\parallel : \mathbf{P} \times \mathbf{P} \to \mathbf{P}$;
- the binary *left merge* operator $⌊⌊ : \mathbf{P} \times \mathbf{P} \to \mathbf{P}$;
- the binary *communication merge* operator $| : \mathbf{P} \times \mathbf{P} \to \mathbf{P}$;
- for each $H \subseteq \mathsf{A}$, the unary *encapsulation* operator $\partial_H : \mathbf{P} \to \mathbf{P}$.

We use infix notation for the additional binary operators as well.

The constants and operators of $\mathrm{ACP}^{\mathrm{c}}$ to build terms of sort $\mathbf{P}$ are the constants and operators of ACP and additionally the guarded command operator.

Let p and q be closed terms of sort $\mathbf{P}$. Intuitively, the additional operators can be explained as follows:

- $p \parallel q$ behaves as the process that proceeds with p and q in parallel;
- $p ⌊⌊ q$ behaves the same as $p \parallel q$, except that it starts with performing an action of p;
- $p \mid q$ behaves the same as $p \parallel q$, except that it starts with performing an action of p and an action of q synchronously;
- $\partial_H(p)$ behaves the same as p, except that it does not perform actions in H.

The axioms of $\mathrm{ACP}^{\mathrm{c}}$ are the axioms of $\mathrm{BPA}_\delta^{\mathrm{c}}$ and the additional axioms given in Table 4. CM2–CM3, CM5–CM7, C1–C3 and D1–D2 are actually axiom schemas in which a, b and c stand for arbitrary constants of sort $\mathbf{P}$ (keep in mind that also the deadlock constant belongs to the constants of sort $\mathbf{P}$). In D1–D4, H stands for an arbitrary subset of A. So, D3 and D4 are axiom schemas as well. Axioms A1–A7, CM1–CM9, C1–C3 and D1–D4 are the axioms of ACP. So $\mathrm{ACP}^{\mathrm{c}}$ imports the axioms of both BA and ACP.

A well-known subtheory of ACP is PA, ACP without communication. Likewise, we have a subtheory of $\mathrm{ACP}^{\mathrm{c}}$, to wit PA^{c}. The theory PA^{c} is $\mathrm{ACP}^{\mathrm{c}}$ without the communication merge operator, without axioms CM5–CM9 and C1–C3, and with axiom CM1 replaced by $x \parallel y = x ⌊⌊ y + y ⌊⌊ x$ (M1). In other words, the possibility that actions are performed synchronously is not covered by PA^{c}.

Table 4
Additional axioms for ACP$^{\mathrm{c}}$ $(a, b, c \in \mathsf{A}_\delta)$

$x \parallel y = x \mathbin{\lfloor\!\lfloor} y + y \mathbin{\lfloor\!\lfloor} x + x \mid y$	CM1	$\partial_H(a) = a$	if $a \notin H$	D1
$a \mathbin{\lfloor\!\lfloor} x = a \cdot x$	CM2	$\partial_H(a) = \delta$	if $a \in H$	D2
$a \cdot x \mathbin{\lfloor\!\lfloor} y = a \cdot (x \parallel y)$	CM3	$\partial_H(x + y) = \partial_H(x) + \partial_H(y)$		D3
$(x + y) \mathbin{\lfloor\!\lfloor} z = x \mathbin{\lfloor\!\lfloor} z + y \mathbin{\lfloor\!\lfloor} z$	CM4	$\partial_H(x \cdot y) = \partial_H(x) \cdot \partial_H(y)$		D4
$a \cdot x \mid b = (a \mid b) \cdot x$	CM5			
$a \mid b \cdot x = (a \mid b) \cdot x$	CM6	$(\phi :\to x) \mathbin{\lfloor\!\lfloor} y = \phi :\to (x \mathbin{\lfloor\!\lfloor} y)$		GC8
$a \cdot x \mid b \cdot y = (a \mid b) \cdot (x \parallel y)$	CM7	$(\phi :\to x) \mid y = \phi :\to (x \mid y)$		GC9
$(x + y) \mid z = x \mid z + y \mid z$	CM8	$x \mid (\phi :\to y) = \phi :\to (x \mid y)$		GC10
$x \mid (y + z) = x \mid y + x \mid z$	CM9	$\partial_H(\phi :\to x) = \phi :\to \partial_H(x)$		GC11
$a \mid b = b \mid a$	C1			
$(a \mid b) \mid c = a \mid (b \mid c)$	C2			
$\delta \mid a = \delta$	C3			

The structural operational semantics of ACP$^{\mathrm{c}}$ is described by the transition rules for BPA$^{\mathrm{c}}_\delta$ and the additional transition rules given in Table 5.

7 Full Splitting Bisimulation Models of ACP$^{\mathrm{c}}$

In this section, we expand the full splitting bisimulation models of BPA$^{\mathrm{c}}_\delta$ to ACP$^{\mathrm{c}}$. We will use the abbreviation $s \xrightarrow{a} s' \wr s''$ for $s \xrightarrow{a} s' \lor (s \xrightarrow{a} \surd \land s' = s'')$. Usually, s'' is a state that takes the place of s' in the case of termination. This is useful where termination has to be turned into a state, as with parallel composition of conditional transition systems.

First of all, we associate with each additional operator f of ACP$^{\mathrm{c}}$ an operation $\widehat{f}$ on $\mathbb{CTS}_\kappa$ as follows.

- Let $T_i = (S_i, \to_i, \to\surd_i, s^0_i) \in \mathbb{CTS}_\kappa$ for $i = 1, 2$. Then

$$T_1 \mathbin{\widehat{\parallel}} T_2 = (S, \to, \to\surd, s^0)\ ,$$

Table 5
Additional transition rules for ACP$^{\mathrm{c}}$

$$\frac{x \xrightarrow{[\phi]\,a} \surd}{x \parallel y \xrightarrow{[\phi]\,a} y} \qquad \frac{y \xrightarrow{[\phi]\,a} \surd}{x \parallel y \xrightarrow{[\phi]\,a} x} \qquad \frac{x \xrightarrow{[\phi]\,a} x'}{x \parallel y \xrightarrow{[\phi]\,a} x' \parallel y} \qquad \frac{y \xrightarrow{[\phi]\,a} y'}{x \parallel y \xrightarrow{[\phi]\,a} x \parallel y'}$$

$$\frac{x \xrightarrow{[\phi]\,a} \surd,\ y \xrightarrow{[\psi]\,b} \surd}{x \parallel y \xrightarrow{[\phi \sqcap \psi]\,c} \surd}\ a \mid b = c,\ \phi \sqcap \psi \neq \bot$$

$$\frac{x \xrightarrow{[\phi]\,a} \surd,\ y \xrightarrow{[\psi]\,b} y'}{x \parallel y \xrightarrow{[\phi \sqcap \psi]\,c} y'}\ a \mid b = c,\ \phi \sqcap \psi \neq \bot$$

$$\frac{x \xrightarrow{[\phi]\,a} x',\ y \xrightarrow{[\psi]\,b} \surd}{x \parallel y \xrightarrow{[\phi \sqcap \psi]\,c} x'}\ a \mid b = c,\ \phi \sqcap \psi \neq \bot$$

$$\frac{x \xrightarrow{[\phi]\,a} x',\ y \xrightarrow{[\psi]\,b} y'}{x \parallel y \xrightarrow{[\phi \sqcap \psi]\,c} x' \parallel y'}\ a \mid b = c,\ \phi \sqcap \psi \neq \bot$$

$$\frac{x \xrightarrow{[\phi]\,a} \surd}{x \mathbin{\lfloor\!\!\lfloor} y \xrightarrow{[\phi]\,a} y} \qquad \frac{x \xrightarrow{[\phi]\,a} x'}{x \mathbin{\lfloor\!\!\lfloor} y \xrightarrow{[\phi]\,a} x' \parallel y}$$

$$\frac{x \xrightarrow{[\phi]\,a} \surd,\ y \xrightarrow{[\psi]\,b} \surd}{x \mid y \xrightarrow{[\phi \sqcap \psi]\,c} \surd}\ a \mid b = c,\ \phi \sqcap \psi \neq \bot$$

$$\frac{x \xrightarrow{[\phi]\,a} \surd,\ y \xrightarrow{[\psi]\,b} y'}{x \mid y \xrightarrow{[\phi \sqcap \psi]\,c} y'}\ a \mid b = c,\ \phi \sqcap \psi \neq \bot$$

$$\frac{x \xrightarrow{[\phi]\,a} x',\ y \xrightarrow{[\psi]\,b} \surd}{x \mid y \xrightarrow{[\phi \sqcap \psi]\,c} x'}\ a \mid b = c,\ \phi \sqcap \psi \neq \bot$$

$$\frac{x \xrightarrow{[\phi]\,a} x',\ y \xrightarrow{[\psi]\,b} y'}{x \mid y \xrightarrow{[\phi \sqcap \psi]\,c} x' \parallel y'}\ a \mid b = c,\ \phi \sqcap \psi \neq \bot$$

$$\frac{x \xrightarrow{[\phi]\,a} \surd}{\partial_H(x) \xrightarrow{[\phi]\,a} \surd}\ a \notin H \qquad \frac{x \xrightarrow{[\phi]\,a} x'}{\partial_H(x) \xrightarrow{[\phi]\,a} \partial_H(x')}\ a \notin H$$

where

$$\begin{aligned}
s^0 &= (s_1^0, s_2^0)\ , \\
s^{\surd} &= \mathrm{ch}_\kappa(\mathcal{S}_\kappa \setminus (S_1 \cup S_2))\ , \\
S &= ((S_1 \cup \{s^{\surd}\}) \times (S_2 \cup \{s^{\surd}\})) \setminus \{(s^{\surd}, s^{\surd})\}\ ,
\end{aligned}$$

and for every $(\alpha, a) \in \mathcal{C}_\kappa^- \times \mathsf{A}$:

$$\begin{aligned}
\xrightarrow{(\alpha,a)} \;&= \Big\{((s_1,s_2),(s_1',s_2)) \;\Big|\; (s_1',s_2) \in S \wedge s_1 \xrightarrow{[\alpha]\,a}_1 s_1' \wr s^{\checkmark}\Big\} \\
&\cup \Big\{((s_1,s_2),(s_1,s_2')) \;\Big|\; (s_1,s_2') \in S \wedge s_2 \xrightarrow{[\alpha]\,a}_2 s_2' \wr s^{\checkmark}\Big\} \\
&\cup \Big\{((s_1,s_2),(s_1',s_2')) \;\Big|\; (s_1',s_2') \in S \wedge {} \\
&\qquad \bigvee_{\alpha',\beta' \in \mathcal{C}_\kappa^-,\, a',b' \in \mathsf{A}} \Big(s_1 \xrightarrow{[\alpha']\,a'}_1 s_1' \wr s^{\checkmark} \wedge s_2 \xrightarrow{[\beta']\,b'}_2 s_2' \wr s^{\checkmark} \wedge {} \\
&\qquad\qquad \alpha' \sqcap \beta' = \alpha \wedge a' \mid b' = a\Big)\Big\} \;,
\end{aligned}$$

$$\begin{aligned}
\xrightarrow{(\alpha,a)}\surd \;&= \Big\{(s_1,s^{\checkmark}) \;\Big|\; s_1 \xrightarrow{[\alpha]\,a} \surd_1\Big\} \\
&\cup \Big\{(s^{\checkmark},s_2) \;\Big|\; s_2 \xrightarrow{[\alpha]\,a} \surd_2\Big\} \\
&\cup \Big\{(s_1,s_2) \;\Big| \\
&\qquad \bigvee_{\alpha',\beta' \in \mathcal{C}_\kappa^-,\, a',b' \in \mathsf{A}} \Big(s_1 \xrightarrow{[\alpha']\,a'} \surd_1 \wedge s_2 \xrightarrow{[\beta']\,b'} \surd_2 \wedge {} \\
&\qquad\qquad \alpha' \sqcap \beta' = \alpha \wedge a' \mid b' = a\Big)\Big\} \;.
\end{aligned}$$

- Let $T_i = (S_i, \rightarrow_i, \rightarrow\surd_i, s_i^0) \in \mathbb{CTS}_\kappa$ for $i = 1, 2$. Suppose that $T_1 \mathbin{\widehat{\|}} T_2 = (S, \rightarrow, \rightarrow\surd, s^0)$ where $S = ((S_1 \cup \{s^{\checkmark}\}) \times (S_2 \cup \{s^{\checkmark}\})) \setminus \{(s^{\checkmark}, s^{\checkmark})\}$ and $s^{\checkmark} = \mathrm{ch}_\kappa(\mathcal{S}_\kappa \setminus (S_1 \cup S_2))$. Then

  $$T_1 \mathbin{\widehat{\lfloor\!\lfloor}} T_2 = \Gamma(S', \rightarrow', \rightarrow\surd, s^{0\prime}) \;,$$

  where

  $$\begin{aligned}
  s^{0\prime} &= \mathrm{ch}_\kappa(\mathcal{S}_\kappa \setminus S) \;, \\
  S' &= \{s^{0\prime}\} \cup S \;,
  \end{aligned}$$

  and for every $(\alpha, a) \in \mathcal{C}_\kappa^- \times \mathsf{A}$:

  $$\xrightarrow{(\alpha,a)}{}' = \Big\{(s^{0\prime}, (s, s_2^0)) \;\Big|\; s_1^0 \xrightarrow{[\alpha]\,a}_1 s \wr s^{\checkmark}\Big\} \cup \xrightarrow{(\alpha,a)} \;.$$

- Let $T_i = (S_i, \rightarrow_i, \rightarrow\surd_i, s_i^0) \in \mathbb{CTS}_\kappa$ for $i = 1, 2$. Suppose that $T_1 \mathbin{\widehat{\|}} T_2 = (S, \rightarrow, \rightarrow\surd, s^0)$ where $S = ((S_1 \cup \{s^{\checkmark}\}) \times (S_2 \cup \{s^{\checkmark}\})) \setminus \{(s^{\checkmark}, s^{\checkmark})\}$ and $s^{\checkmark} = \mathrm{ch}_\kappa(\mathcal{S}_\kappa \setminus (S_1 \cup S_2))$. Then

  $$T_1 \mathbin{\widehat{|}} T_2 = \Gamma(S', \rightarrow', \rightarrow\surd', s^{0\prime}) \;,$$

  where

  $$\begin{aligned}
  s^{0\prime} &= \mathrm{ch}_\kappa(\mathcal{S}_\kappa \setminus S) \;, \\
  S' &= \{s^{0\prime}\} \cup S \;,
  \end{aligned}$$

and for every $(\alpha, a) \in \mathcal{C}_\kappa^- \times \mathsf{A}$:

$$\begin{aligned}
\xrightarrow{(\alpha,a)}{}' \;&= \Big\{(s^{0\prime}, (s_1, s_2)) \;\Big|\; (s_1, s_2) \in S \wedge \\
&\qquad \bigvee_{\alpha',\beta' \in \mathcal{C}_\kappa^-,\, a',b' \in \mathsf{A}} \Big(s_1^0 \xrightarrow{[\alpha']\, a'}_1 s_1 \wr s^{\checkmark} \wedge s_2^0 \xrightarrow{[\beta']\, b'}_2 s_2 \wr s^{\checkmark} \wedge \\
&\qquad\qquad \alpha' \sqcap \beta' = \alpha \wedge a' \mid b' = a\Big)\Big\} \\
&\quad \cup \xrightarrow{(\alpha,a)} , \\
\xrightarrow{(\alpha,a)} \surd' &= \Big\{s^{0\prime} \;\Big| \\
&\qquad \bigvee_{\alpha',\beta' \in \mathcal{C}_\kappa^-,\, a',b' \in \mathsf{A}} \Big(s_1^0 \xrightarrow{[\alpha']\, a'} \surd_1 \wedge s_2^0 \xrightarrow{[\beta']\, b'} \surd_2 \wedge \\
&\qquad\qquad \alpha' \sqcap \beta' = \alpha \wedge a' \mid b' = a\Big)\Big\} \\
&\quad \cup \xrightarrow{(\alpha,a)} \surd .
\end{aligned}$$

- Let $T = (S, \rightarrow, \rightarrow\surd, s^0) \in \mathbb{CTS}_\kappa$. Then

  $$\widehat{\partial_H}(T) = \Gamma(S, \rightarrow', \rightarrow\surd', s^0) ,$$

  where for every $(\alpha, a) \in \mathcal{C}_\kappa^- \times (\mathsf{A} \setminus H)$:

  $$\xrightarrow{(\alpha,a)}{}' \;= \xrightarrow{(\alpha,a)} ,$$

  $$\xrightarrow{(\alpha,a)} \surd' = \xrightarrow{(\alpha,a)} \surd ,$$

  and for every $(\alpha, a) \in \mathcal{C}_\kappa^- \times H$:

  $$\xrightarrow{(\alpha,a)}{}' \;= \emptyset ,$$

  $$\xrightarrow{(\alpha,a)} \surd' = \emptyset .$$

We can easily show that splitting bisimilarity is a congruence with respect to parallel composition, left merge, communication merge and encapsulation.

Proposition 8 (Congruence) *Let κ be an infinite cardinal. Then for all $T_1, T_2, T_1', T_2' \in \mathbb{CTS}_\kappa$, $T_1 \underline{\leftrightarrow} T_1'$ and $T_2 \underline{\leftrightarrow} T_2'$ imply $T_1 \mathbin{\widehat{\|}} T_2 \underline{\leftrightarrow} T_1' \mathbin{\widehat{\|}} T_2'$, $T_1 \mathbin{\widehat{\lfloor\!\lfloor}} T_2 \underline{\leftrightarrow} T_1' \mathbin{\widehat{\lfloor\!\lfloor}} T_2'$, $T_1 \mathbin{\widehat{|}} T_2 \underline{\leftrightarrow} T_1' \mathbin{\widehat{|}} T_2'$ and $\widehat{\partial_H}(T_1) \underline{\leftrightarrow} \widehat{\partial_H}(T_1')$.*

PROOF. Let $T_i = (S_i, \rightarrow_i, \rightarrow\surd_i, s_i^0)$ and $T_i' = (S_i', \rightarrow_i', \rightarrow\surd_i', s_i^{0\prime})$ for $i = 1, 2$. Let R_1 and R_2 be splitting bisimulations witnessing $T_1 \underline{\leftrightarrow} T_1'$ and $T_2 \underline{\leftrightarrow} T_2'$, respectively. Then we construct relations $R_{\widehat{\|}}$, $R_{\widehat{\lfloor\!\lfloor}}$, $R_{\widehat{|}}$ and $R_{\widehat{\partial_H}}$ as follows:

- $R_{\widehat{\|}} = \{((s_1, s_2), (s_1', s_2')) \in S \times S' \mid (s_1, s_1') \in R_1 \cup R^{\checkmark}, (s_2, s_2') \in R_2 \cup R^{\checkmark}\}$, where S and S' are the sets of states of $T_1 \mathbin{\widehat{\|}} T_2$ and $T_1' \mathbin{\widehat{\|}} T_2'$, respectively, and

$R^{\surd} = \{(\mathrm{ch}_\kappa(\mathcal{S}_\kappa \setminus (S_1 \cup S_2)), \mathrm{ch}_\kappa(\mathcal{S}_\kappa \setminus (S'_1 \cup S'_2)))\}$;

- $R_{\widehat{\lfloor\!\lfloor}} = (\{(s^0, s^{0\prime})\} \cup R_{\widehat{\|}}) \cap (S \times S')$, where S and S' are the sets of states of $T_1 \mathbin{\widehat{\lfloor\!\lfloor}} T_2$ and $T'_1 \mathbin{\widehat{\lfloor\!\lfloor}} T'_2$, respectively, and s^0 and $s^{0\prime}$ are the initial states of $T_1 \mathbin{\widehat{\lfloor\!\lfloor}} T_2$ and $T'_1 \mathbin{\widehat{\lfloor\!\lfloor}} T'_2$, respectively;
- $R_{\widehat{|}} = (\{(s^0, s^{0\prime})\} \cup R_{\widehat{\|}}) \cap (S \times S')$, where S and S' are the sets of states of $T_1 \mathbin{\widehat{|}} T_2$ and $T'_1 \mathbin{\widehat{|}} T'_2$, respectively, and s^0 and $s^{0\prime}$ are the initial states of $T_1 \mathbin{\widehat{|}} T_2$ and $T'_1 \mathbin{\widehat{|}} T'_2$, respectively;
- $R_{\widehat{\partial_H}} = R_1 \cap (S \times S')$, where S and S' are the sets of states of $\widehat{\partial_H}(T_1)$ and $\widehat{\partial_H}(T'_1)$, respectively.

Given the definitions of parallel composition, left merge, communication merge and encapsulation, it is easy to see that $R_{\widehat{\|}}$, $R_{\widehat{\lfloor\!\lfloor}}$, $R_{\widehat{|}}$ and $R_{\widehat{\partial_H}}$ are splitting bisimulations witnessing $T_1 \mathbin{\widehat{\|}} T_2 \leftrightarroweq T'_1 \mathbin{\widehat{\|}} T'_2$, $T_1 \mathbin{\widehat{\lfloor\!\lfloor}} T_2 \leftrightarroweq T'_1 \mathbin{\widehat{\lfloor\!\lfloor}} T'_2$, $T_1 \mathbin{\widehat{|}} T_2 \leftrightarroweq T'_1 \mathbin{\widehat{|}} T'_2$ and $\widehat{\partial_H}(T_1) \leftrightarroweq \widehat{\partial_H}(T'_1)$, respectively. □

The *full splitting bisimulation models* $\mathfrak{P}^{\mathrm{c}\prime}_\kappa$ of ACP$^\mathrm{c}$, one for each infinite cardinal κ, are the expansions of the full splitting bisimulation models $\mathfrak{P}^{\mathrm{c}}_\kappa$ of BPA$^\mathrm{c}_\delta$ with an operation $\tilde{f}$ on $\mathbb{CTS}_\kappa/{\leftrightarroweq}$ for each additional operator f of ACP$^\mathrm{c}$. Those additional operations are defined as follows:

$$
\begin{aligned}
[\,T_1\,]_{\leftrightarroweq} \mathbin{\widetilde{\|}} [\,T_2\,]_{\leftrightarroweq} &= [\,T_1 \mathbin{\widehat{\|}} T_2\,]_{\leftrightarroweq}\ ,\\
[\,T_1\,]_{\leftrightarroweq} \mathbin{\widetilde{\lfloor\!\lfloor}} [\,T_2\,]_{\leftrightarroweq} &= [\,T_1 \mathbin{\widehat{\lfloor\!\lfloor}} T_2\,]_{\leftrightarroweq}\ ,\\
[\,T_1\,]_{\leftrightarroweq} \mathbin{\widetilde{|}} [\,T_2\,]_{\leftrightarroweq} &= [\,T_1 \mathbin{\widehat{|}} T_2\,]_{\leftrightarroweq}\ ,\\
\widetilde{\partial_H}([\,T_1\,]_{\leftrightarroweq}) \qquad &= [\,\widehat{\partial_H}(T_1)\,]_{\leftrightarroweq}\ .
\end{aligned}
$$

Parallel composition, left merge, communication merge and encapsulation on $\mathbb{CTS}_\kappa/{\leftrightarroweq}$ are well-defined because $\leftrightarroweq$ is a congruence with respect to the corresponding operations on $\mathbb{CTS}_\kappa$.

The structures $\mathfrak{P}^{\mathrm{c}\prime}_\kappa$ are models of ACP$^\mathrm{c}$.

Theorem 9 (Soundness of ACP$^\mathrm{c}$) *For each infinite cardinal κ, we have $\mathfrak{P}^{\mathrm{c}\prime}_\kappa \models \mathrm{ACP}^\mathrm{c}$.*

PROOF. Because $\mathfrak{P}^{\mathrm{c}\prime}_\kappa$ is an expansion of $\mathfrak{P}^{\mathrm{c}}_\kappa$, it is sufficient to show that the additional axioms for ACP$^\mathrm{c}$ are sound. The soundness of all additional axioms follows easily from the definitions of the ingredients of $\mathfrak{P}^{\mathrm{c}\prime}_\kappa$. □

It is easy to see that Theorem 7 goes through for $\mathfrak{P}^{\mathrm{c}\prime}_\kappa$.

Table 6
Axioms for guarded recursion

$\langle X\|E\rangle = \langle t_X\|E\rangle$	if $X = t_X \in E$	RDP
$E \Rightarrow X = \langle X\|E\rangle$	if $X \in \mathrm{V}(E)$	RSP

In this section, the full splitting bisimulation models $\mathfrak{P}^{\mathrm{c}}_{\kappa}$ of $\mathrm{BPA}^{\mathrm{c}}_{\delta}$ have been expanded to obtain the full splitting bisimulation models $\mathfrak{P}^{\mathrm{c}\prime}_{\kappa}$ of $\mathrm{ACP}^{\mathrm{c}}$. Henceforth, we will loosely write $\mathfrak{P}^{\mathrm{c}}_{\kappa}$ for $\mathfrak{P}^{\mathrm{c}\prime}_{\kappa}$. It is always made sure that no confusion between the original model and its expansion may arise.

8 Guarded Recursion

In order to allow for the description of (potentially) non-terminating processes, we add guarded recursion to $\mathrm{ACP}^{\mathrm{c}}$.

A *recursive specification* over $\mathrm{ACP}^{\mathrm{c}}$ is a set of *recursive* equations $E = \{X = t_X \mid X \in V\}$ where V is a set of variables and each t_X is a term of sort $\mathbf{P}$ that only contains variables from V. We write $\mathrm{V}(E)$ for the set of all variables that occur on the left-hand side of an equation in E. A *solution* of a recursive specification E is a set of processes (in some model of $\mathrm{ACP}^{\mathrm{c}}$) $\{P_X \mid X \in \mathrm{V}(E)\}$ such that the equations of E hold if, for all $X \in \mathrm{V}(E)$, X stands for P_X.

Let t be a term of sort $\mathbf{P}$ containing a variable X. We call an occurrence of X in t *guarded* if t has a subterm of the form $a \cdot t'$, where $a \in \mathsf{A}$ and t' a term of sort $\mathbf{P}$, with t' containing this occurrence of X. A recursive specification over $\mathrm{ACP}^{\mathrm{c}}$ is called a *guarded* recursive specification if all occurrences of variables in the right-hand sides of its equations are guarded or it can be rewritten to such a recursive specification using the axioms of $\mathrm{ACP}^{\mathrm{c}}$ and the equations of the recursive specification. We are only interested in models of $\mathrm{ACP}^{\mathrm{c}}$ in which guarded recursive specifications have unique solutions.

For each guarded recursive specification E and each variable $X \in \mathrm{V}(E)$, we introduce a constant of sort $\mathbf{P}$ standing for the unique solution of E for X. This constant is denoted by $\langle X|E\rangle$. We often write X for $\langle X|E\rangle$ if E is clear from the context. In such cases, it should also be clear from the context that we use X as a constant.

We will also use the following notation. Let t be a term of sort $\mathbf{P}$ and E be a guarded recursive specification over $\mathrm{ACP}^{\mathrm{c}}$. Then we write $\langle t|E\rangle$ for t with, for all $X \in \mathrm{V}(E)$, all occurrences of X in t replaced by $\langle X|E\rangle$.

The additional axioms for guarded recursion are the equations given in Table 6. Both RDP and RSP are axiom schemas. A side condition is added to restrict

Table 7
Transition rules for guarded recursion

$$\frac{\langle t_X|E\rangle \xrightarrow{[\phi]\,a} \surd}{\langle X|E\rangle \xrightarrow{[\phi]\,a} \surd}\; X = t_X \in E \qquad \frac{\langle t_X|E\rangle \xrightarrow{[\phi]\,a} x'}{\langle X|E\rangle \xrightarrow{[\phi]\,a} x'}\; X = t_X \in E$$

the variables, terms and guarded recursive specifications for which X, t_X and E stand. The additional axioms for guarded recursion are known as the recursive definition principle (RDP) and the recursive specification principle (RSP). The equations $\langle X|E\rangle = \langle t_X|E\rangle$ for a fixed E express that the constants $\langle X|E\rangle$ make up a solution of E. The conditional equations $E \Rightarrow X = \langle X|E\rangle$ express that this solution is the only one.

The structural operational semantics for the constants $\langle X|E\rangle$ is described by the transition rules given in Table 7.

In the full splitting bisimulation models of ACP$^{\mathrm{c}}$, guarded recursive specifications over ACP$^{\mathrm{c}}$ have unique solutions.

Theorem 10 (Unique solutions in $\mathfrak{P}^{\mathrm{c}}_{\kappa}$) *For each infinite cardinal κ, guarded recursive specifications over* ACP$^{\mathrm{c}}$ *have unique solutions in $\mathfrak{P}^{\mathrm{c}}_{\kappa}$.*

PROOF. In [20], a proof of uniqueness of solutions of guarded recursive specifications in the graph models of ACP$_\tau$ is given. That proof can easily be adapted to the full bisimulation models of ACP introduced in [19]. The proof consists of the following three steps: (i) proving that two transition systems are bisimilar if at least one of them is finitely branching and all their finite projections are bisimilar; (ii) proving, using the result of step (i), that every guarded recursive specification has a solution that is finitely branching; (iii) proving, using the result of step (i), that the solution from step (ii) is bisimilar to any other solution. Steps (ii) and (iii) remain essentially the same in the case of conditional transition systems and splitting bisimilarity. It is straightforward to define a normal form of elements of $\mathbb{CTS}_\kappa$ such that: (a) each element of $\mathbb{CTS}_\kappa$ is splitting bisimilar to its normal form and (b) two elements of $\mathbb{CTS}_\kappa$ are splitting bisimilar iff their normal forms are bisimilar (cf. the last two paragraphs of Section 3). This enables us to adapt step (i) easily to the case of conditional transition systems and splitting bisimilarity as well. □

Thus, the full splitting bisimulation models $\mathfrak{P}^{\mathrm{c}\,\prime\prime}_{\kappa}$ of ACP$^{\mathrm{c}}$ with guarded recursion are simply the expansions of the full splitting bisimulation models $\mathfrak{P}^{\mathrm{c}}_{\kappa}$ of ACP$^{\mathrm{c}}$ obtained by associating with each constant $\langle X|E\rangle$ the unique solution of E for X in the full splitting bisimulation model concerned.

Table 8
Axioms for condition evaluation ($a \in \mathsf{A}_\delta$, $\eta \in \mathsf{C}_{\mathsf{at}}$, $\eta' \in \mathsf{C}_{\mathsf{at}} \cup \{\bot, \top\}$)

$\mathsf{CE}_h(a) = a$		CE1
$\mathsf{CE}_h(a \cdot x) = a \cdot \mathsf{CE}_h(x)$		CE2
$\mathsf{CE}_h(x + y) = \mathsf{CE}_h(x) + \mathsf{CE}_h(y)$		CE3
$\mathsf{CE}_h(\phi :\rightarrow x) = \mathsf{CE}_h(\phi) :\rightarrow \mathsf{CE}_h(x)$		CE4
$\mathsf{CE}_h(\mathsf{CE}_{h'}(x)) = \mathsf{CE}_{h \circ h'}(x)$		CE5
$\mathsf{CE}_h(\bot) = \bot$		CE6
$\mathsf{CE}_h(\top) = \top$		CE7
$\mathsf{CE}_h(\eta) = \eta'$	if $h(\eta) = \eta'$	CE8
$\mathsf{CE}_h(-\phi) = -\mathsf{CE}_h(\phi)$		CE9
$\mathsf{CE}_h(\phi \sqcup \psi) = \mathsf{CE}_h(\phi) \sqcup \mathsf{CE}_h(\psi)$		CE10
$\mathsf{CE}_h(\phi \sqcap \psi) = \mathsf{CE}_h(\phi) \sqcap \mathsf{CE}_h(\psi)$		CE11

9 Evaluation of Conditions

Guarded commands cannot always be eliminated from closed terms of sort **P** because conditions different from both $\bot$ and $\top$ may be involved. The condition evaluation operators introduced below, can be brought into action in such cases. These operators require to fix an infinite cardinal λ. By doing so, full splitting bisimulation models with process domain $\mathbb{CTS}_\kappa/\underline{\leftrightarrow}$ for $\kappa > \lambda$ are excluded.

There are unary λ-*complete condition evaluation* operators $\mathsf{CE}_h : \mathbf{P} \to \mathbf{P}$ and $\mathsf{CE}_h : \mathbf{C} \to \mathbf{C}$ for each λ-complete endomorphisms h of $\mathcal{C}_\lambda$.[7]

These operators can be explained as follows: $\mathsf{CE}_h(p)$ behaves as p with each condition ζ occurring in p replaced according to h. If the image of $\mathcal{C}_\lambda$ under h is $\mathbb{B}$, i.e. the Boolean algebra with domain $\{\bot, \top\}$, then guarded commands can be eliminated from $\mathsf{CE}_h(p)$. In the case where the image of $\mathcal{C}_\lambda$ under h is not $\mathbb{B}$, CE_h can be regarded to evaluate the conditions only partially.

Henceforth, we write $\mathcal{H}_\lambda$ for the set of all λ-complete endomorphisms of $\mathcal{C}_\lambda$.

The additional axioms for CE_h, where $h \in \mathcal{H}_\lambda$, are the axioms given in Table 8.

Example 11 *We return to Example 1, which is concerned with a pedestrian who uses a crossing with traffic lights to cross a road with busy traffic safely.*

[7] For a definition of κ-complete endomorphisms, see e.g. [11].

Table 9

Transition rules for condition evaluation

$$\frac{x \xrightarrow{[\phi]\, a} \surd}{\mathsf{CE}_h(x) \xrightarrow{[h(\phi)]\, a} \surd}\ h(\phi) \neq \bot \qquad \frac{x \xrightarrow{[\phi]\, a} x'}{\mathsf{CE}_h(x) \xrightarrow{[h(\phi)]\, a} \mathsf{CE}_h(x')}\ h(\phi) \neq \bot$$

Recall that the description of the behaviour of the pedestrian given in Example 1 is as follows:

$$\mathit{PED} = \mathit{arrive} \cdot (\mathit{green} :\to \mathit{cross} + \mathit{red} :\to (\mathit{make\_req} \cdot (\mathit{green} :\to \mathit{cross})))\ .$$

Let h_g be such that $h_g(\mathit{green}) = \top$ and $h_g(\mathit{red}) = \bot$; and let h_r be such that $h_r(\mathit{green}) = \bot$ and $h_r(\mathit{red}) = \top$. Then we can derive the following:

$$\mathsf{CE}_{h_g}(\mathit{PED}) = \mathit{arrive} \cdot \mathit{cross}\ ,$$
$$\mathsf{CE}_{h_r}(\mathit{PED}) = \mathit{arrive} \cdot \mathit{make\_req} \cdot \delta\ .$$

So in a world where the traffic light for pedestrians is green he or she will cross the street without making a request for green light; and in a world where the traffic light for pedestrians is red he or she will become completely inactive after making a request for green light. In reality, the request would cause a change from red to green, but the condition evaluation operators CE_h cannot deal with that. We will return to this issue in Example 15.

The structural operational semantics of ACP$^{\mathrm{c}}$ extended with condition evaluation is described by the transition rules for ACP$^{\mathrm{c}}$ and the transition rules given in Table 9.

If λ is a regular infinite cardinal, the elements of $\mathcal{C}_\lambda$ can be used to represent equivalence classes with respect to logical equivalence of the set of all propositions with elements of $\mathsf{C_{at}}$ as propositional variables and with conjunctions and disjunctions of less than λ propositions. We write $\mathcal{P}_\lambda$ for this set of propositions. Suppose that a theory Φ about the set of atomic conditions $\mathsf{C_{at}}$ is given in the shape of a subset of $\mathcal{P}_\lambda$. Then we can associate a λ-complete endomorphism h_Φ with Φ as set out below.

Let λ be a regular infinite cardinal, let $\Phi \subset \mathcal{P}_\lambda$, and let $h_\Phi \in \mathcal{H}_\lambda$ be such that for all $\alpha, \beta \in \mathcal{C}_\lambda$:

$$\Phi \vdash \langle\!\langle h_\Phi(\alpha) \rangle\!\rangle \Leftrightarrow \langle\!\langle \alpha \rangle\!\rangle \quad \text{and} \quad h_\Phi(\alpha) = h_\Phi(\beta) \text{ iff } \Phi \vdash \langle\!\langle \alpha \rangle\!\rangle \Leftrightarrow \langle\!\langle \beta \rangle\!\rangle \qquad (1)$$

where $\langle\!\langle \alpha \rangle\!\rangle$ is a representative of the equivalence class of propositions isomorphic to α. Then we have $h_\Phi(\alpha) = \top$ iff $\langle\!\langle \alpha \rangle\!\rangle$ is derivable from Φ and $h_\Phi(\alpha) = \bot$ iff $\neg\langle\!\langle \alpha \rangle\!\rangle$ is derivable from Φ. The image of $\mathcal{C}_\lambda$ under h_Φ is $\mathbb{B}$ iff Φ

is a complete theory. If Φ is not a complete theory, then h_Φ is not uniquely determined by (1). However, the images of $\mathcal{C}_\lambda$ under the different endomorphisms satisfying (1) are isomorphic subalgebras of $\mathcal{C}_\lambda$. Moreover, if both h and h' satisfy (1), then $\Phi \vdash \langle\!\langle h(\alpha) \rangle\!\rangle \Leftrightarrow \langle\!\langle h'(\alpha) \rangle\!\rangle$ for all $\alpha \in \mathcal{C}_\lambda$.

Example 12 *In Example 11, where $\mathsf{C}_{\mathsf{at}} = \{green, red\}$, we introduced the λ-complete endomorphisms h_g and h_r such that $h_g(green) = \top$, $h_g(red) = \bot$, $h_r(green) = \bot$ and $h_r(red) = \top$. Let $\Phi = \{green, \neg red\}$. In the terminology of Example 11, Φ is a theory about the world in which the traffic light for pedestrians is green. Because Φ is a complete theory, h_Φ is uniquely determined by (1). Indeed, we have $h_\Phi = h_g$. Note that h_Φ would not be uniquely determined if the restriction $\Phi \vdash \langle\!\langle h_\Phi(\alpha) \rangle\!\rangle \Leftrightarrow \langle\!\langle \alpha \rangle\!\rangle$ had been left out from (1). In that case, both h_g and h_r could be taken as h_Φ.*

Below, we show that condition evaluation on the basis of a complete theory can be viewed as substitution on the basis of the theory. That leads us to the use of the following convention: for $\alpha \in \mathcal{C}$, $\underline{\alpha}$ stands for an arbitrary closed term of sort $\mathbf{C}$ of which the value in $\mathcal{C}$ is α.

Proposition 13 (Condition evaluation on the basis of a theory) *Assume that λ is a regular infinite cardinal. Let $\Phi \subset \mathcal{P}_\lambda$ be a complete theory and let p be a closed term of sort $\mathbf{P}$. Then $\mathsf{CE}_{h_\Phi}(p) = p'$ where p' is p with, for all $\alpha \in \mathcal{C}$, in all subterms of the form $\underline{\alpha} :\to q$, $\underline{\alpha}$ replaced by $\top$ if $\Phi \vdash \langle\!\langle \alpha \rangle\!\rangle$ and $\underline{\alpha}$ replaced by $\bot$ if $\Phi \vdash \neg\langle\!\langle \alpha \rangle\!\rangle$.*

PROOF. This result follows immediately from the definition of h_Φ and the distributivity of CE_{h_Φ} over all operators of ACP$^{\mathrm{c}}$. $\square$

In μCRL [21], an extension of ACP which includes conditional expressions, we find a formalization of the substitution-based alternative for CE_{h_Φ}.

The substitution-based alternative works properly because condition evaluation by means of a λ-complete condition evaluation operator is not dependent on process behaviour. Hence, the result of condition evaluation is globally valid. Below, we will generalize the condition evaluation operators introduced above in such a way that condition evaluation may be dependent on process behaviour. In that case, the result of condition evaluation is in general not globally valid.

Remark 14 *Assume that λ is a regular infinite cardinal. Let $h \in \mathcal{H}_\lambda$. Then h induces a theory $\Phi \subset \mathcal{P}_\lambda$ such that $h = h_\Phi$, viz. the theory Φ defined by*

$$\Phi = \{\langle\!\langle h(\alpha) \rangle\!\rangle \Leftrightarrow \langle\!\langle \alpha \rangle\!\rangle \mid \alpha \in \mathcal{C}_\lambda\} \cup \{\langle\!\langle \alpha \rangle\!\rangle \Leftrightarrow \langle\!\langle \beta \rangle\!\rangle \mid h(\alpha) = h(\beta)\} \; .$$

Table 10
Axioms for generalized condition evaluation ($a \in \mathsf{A}_\delta$)

$\mathsf{GCE}_h(a) = a$	GCE1
$\mathsf{GCE}_h(a \cdot x) = a \cdot \mathsf{GCE}_{\mathsf{eff}(a,h)}(x)$	GCE2
$\mathsf{GCE}_h(x + y) = \mathsf{GCE}_h(x) + \mathsf{GCE}_h(y)$	GCE3
$\mathsf{GCE}_h(\phi :\rightarrow x) = \mathsf{CE}_h(\phi) :\rightarrow \mathsf{GCE}_h(x)$	GCE4

Consequently, if λ is a regular infinite cardinal, condition evaluation by means of the λ-complete condition evaluation operators introduced above is always condition evaluation of which the result can be determined from a set of propositions. We will return to this observation in Section 11.

We proceed with generalizing the condition evaluation operators introduced above. It is assumed that a fixed but arbitrary function $\mathsf{eff} : \mathsf{A} \times \mathcal{H}_\lambda \rightarrow \mathcal{H}_\lambda$ has been given.

There is a unary *generalized λ-complete condition evaluation* operator $\mathsf{GCE}_h : \mathbf{P} \rightarrow \mathbf{P}$ for each $h \in \mathcal{H}_\lambda$; and there is again the unary operator $\mathsf{CE}_h : \mathbf{C} \rightarrow \mathbf{C}$ for each $h \in \mathcal{H}_\lambda$.

The λ-complete generalized condition evaluation operator GCE_h allows, given the function eff, to evaluate conditions dependent on process behaviour. The function eff gives, for each action a and λ-complete endomorphism h, the λ-complete endomorphism h' that represents the changed results of condition evaluation due to performing a. The function eff is extended to A_δ such that $\mathsf{eff}(\delta, h) = h$ for all $h \in \mathcal{H}_\lambda$.

The additional axioms for GCE_h, where $h \in \mathcal{H}_\lambda$, are the axioms given in Table 10 and axioms CE6–CE11 from Table 8.

Example 15 *We return to Example 1, which is concerned with a pedestrian who uses a crossing with traffic lights to cross a road with busy traffic safely. In Example 11, we illustrated that the condition evaluation operators CE_h cannot deal with the change from red light to green light caused by a request for green light. Here, we illustrate that the generalized condition evaluation operators GCE_h can deal with such a change. Let h_g and h_r be as in Example 11; and let eff be such that $\mathsf{eff}(make\_req, h_r) = h_g$ and $\mathsf{eff}(a, h) = h$ otherwise. Then we can derive the following:*

$$\mathsf{GCE}_{h_g}(PED) = arrive \cdot cross \;,$$
$$\mathsf{GCE}_{h_r}(PED) = arrive \cdot make\_req \cdot cross \;.$$

The change from red light to green light is due to interaction between the pedes-

Table 11
Transition rules for generalized condition evaluation

$\dfrac{x \xrightarrow{[\phi]\,a} \surd}{\mathsf{GCE}_h(x) \xrightarrow{[h(\phi)]\,a} \surd}\; h(\phi) \neq \bot$	$\dfrac{x \xrightarrow{[\phi]\,a} x'}{\mathsf{GCE}_h(x) \xrightarrow{[h(\phi)]\,a} \mathsf{GCE}_{\mathsf{eff}(a,h)}(x')}\; h(\phi) \neq \bot$

trian and the traffic lights. It is clear that this interaction is poorly represented by a generalized condition evaluation operator. We will return to this issue in Example 18.

The structural operational semantics of ACP$^{\mathrm{c}}$ extended with generalized condition evaluation is described by the transition rules for ACP$^{\mathrm{c}}$ and the transition rules given in Table 11.

We can add both the λ-complete condition evaluation operators and the generalized λ-complete condition evaluation operators to ACP$^{\mathrm{c}}$. However, Proposition 16 stated below makes it clear that the latter operators supersede the former operators.

The full splitting bisimulation models of ACP$^{\mathrm{c}}$ with condition evaluation and/or generalized condition evaluation are simply the expansions of the full splitting bisimulation models $\mathfrak{P}^{\mathrm{c}}_\kappa$ of ACP$^{\mathrm{c}}$, for infinite cardinals $\kappa \leq \lambda$, obtained by associating with each operator CE_h and/or GCE_h the corresponding re-labeling operation on conditional transition systems. As mentioned before, full splitting bisimulation models with process domain $\mathbb{CTS}_\kappa/\underline{\leftrightarrow}$ for $\kappa > \lambda$ are excluded.

We write $\mathfrak{P}^{\mathrm{ce}}_\kappa$ for the expansion of $\mathfrak{P}^{\mathrm{c}}_\kappa$ for the λ-complete condition evaluation operators and the generalized λ-complete condition evaluation operators.

As their name suggests, the generalized λ-complete condition evaluation operators are generalizations of the λ-complete condition evaluation operators.

Proposition 16 (Generalization) *We can fix the function* eff *such that* $\mathsf{GCE}_h(x) = \mathsf{CE}_h(x)$ *holds for all* $h \in \mathcal{H}_\lambda$ *in all full splitting bisimulation models* $\mathfrak{P}^{\mathrm{ce}}_\kappa$ *with* $\kappa \leq \lambda$.

PROOF. Clearly, if $\mathsf{eff}(a, h') = h'$ for all $a \in \mathsf{A}$ and $h' \in \mathcal{H}_\lambda$, then $\mathsf{GCE}_h(x) = \mathsf{CE}_h(x)$ holds for all $h \in \mathcal{H}_\lambda$ in all full splitting bisimulation models $\mathfrak{P}^{\mathrm{ce}}_\kappa$ with $\kappa \leq \lambda$. $\square$

The λ-complete state operators that are added to ACP$^{\mathrm{c}}$ in Section 10 are in their turn generalizations of the generalized λ-complete condition evaluation operators.

Note that the equation $\mathsf{CE}_h(\mathsf{CE}_{h'}(x)) = \mathsf{CE}_{h \circ h'}(x)$ is an axiom, but the equation $\mathsf{GCE}_h(\mathsf{GCE}_{h'}(x)) = \mathsf{GCE}_{h \circ h'}(x)$ is not an axiom. The latter equation is only valid in the full splitting bisimulation models $\mathfrak{P}^{\mathrm{ce}}_{\kappa}$ if eff satisfies $\mathsf{eff}(a, h \circ h') = \mathsf{eff}(a, h) \circ \mathsf{eff}(a, h')$ for all $a \in \mathsf{A}$ and $h, h' \in \mathcal{H}_\lambda$.

We come back to the λ-complete condition evaluation operators CE_h for $h \in \mathcal{H}_\lambda$. The image of $\mathcal{C}_\lambda$ under the λ-complete endomorphism h is a subalgebra of $\mathcal{C}_\lambda$ that is λ-complete too. For that reason, we could have used λ-complete homomorphisms to subalgebras that are λ-complete instead of λ-complete endomorphisms. It would go beyond the models of the theory developed so far to generalize this in such a way that λ-complete homomorphisms to λ-complete Boolean algebras different from subalgebras of $\mathcal{C}_\lambda$ are also included.

However, in the case where we consider λ-complete homomorphisms between free λ-complete Boolean algebras over different sets of generators, we can relate the models for different choices for C_{at}.

Let C and C' be different choices for C_{at},[8] and let $\mathfrak{P}^{\mathrm{c}}_{\kappa}(C)$ and $\mathfrak{P}^{\mathrm{c}}_{\kappa}(C')$, for $\kappa \leq \lambda$, be the full splitting bisimulation models $\mathfrak{P}^{\mathrm{c}}_{\kappa}$ of ACP$^{\mathrm{c}}$ for the different choices for C_{at}. Moreover, let h be a λ-complete homomorphism from the free λ-complete Boolean algebra over C to the free λ-complete Boolean algebra over C'. Then h can be extended to a homomorphism h^* from $\mathfrak{P}^{\mathrm{c}}_{\kappa}(C)$ to $\mathfrak{P}^{\mathrm{c}}_{\kappa}(C')$. This homomorphism is defined by

$$h^*([\,(S, \rightarrow, \rightarrow\surd, s^0)\,]_{\Leftrightarrow}) = [\,\Gamma(S, \rightarrow', \rightarrow\surd', s^0)\,]_{\Leftrightarrow}\ ,$$

where for every $(\alpha, a) \in \mathcal{C}^-_\kappa \times \mathsf{A}$:

$$\begin{aligned}
\xrightarrow{(\alpha,a)}{}' \; &= \left\{(s, s') \;\middle|\; \exists \beta \bullet \left(s \xrightarrow{[\beta]\,a} s' \land \alpha = h(\beta)\right)\right\}\ , \\
\xrightarrow{(\alpha,a)}\surd' &= \left\{s \;\middle|\; \exists \beta \bullet \left(s \xrightarrow{[\beta]\,a} \surd \land \alpha = h(\beta)\right)\right\}\ .
\end{aligned}$$

It is easy to see that h^* is well-defined and a homomorphism indeed.

Thus, a λ-complete homomorphism between free λ-complete Boolean algebras over different sets of generators can be used to translate conditions throughout a full splitting bisimulation model for one choice of C_{at} in such a way that a full splitting bisimulation model for a different choice of C_{at} is obtained.

[8] The interesting cases are those where the cardinalities of C and C' are different. Otherwise, the homomorphisms are isomorphisms.

10 State Operators

The state operators make it easy to represent the execution of a process in a state. The basic idea is that the execution of an action in a state has effect on the state, i.e. it causes a change of state. Besides, there is an action left when an action is executed in a state. The operators introduced here generalize the state operators added to ACP in [22]. The main difference with those operators is that guarded commands are taken into account. As in the case of the condition evaluation operators and the generalized condition evaluation operators, these state operators require to fix an infinite cardinal λ. By doing so, full splitting bisimulation models with process domain $\mathbb{CTS}_{\kappa}/{\Leftrightarrow}$ for $\kappa > \lambda$ are excluded.

It is assumed that a fixed but arbitrary set S of *states* has been given, together with functions $\mathsf{act} : \mathsf{A} \times S \to \mathsf{A}_{\delta}$, $\mathsf{eff} : \mathsf{A} \times S \to S$ and $\mathsf{eval} : \mathcal{C}_{\lambda} \times S \to \mathcal{C}_{\lambda}$, where, for each $s \in S$, the function $h_s : \mathcal{C}_{\lambda} \to \mathcal{C}_{\lambda}$ defined by $h_s(\alpha) = \mathsf{eval}(\alpha, s)$ is a λ-complete endomorphism of $\mathcal{C}_{\lambda}$.

There are unary *λ-complete state operators* $\lambda_s : \mathbf{P} \to \mathbf{P}$ and $\lambda_s : \mathbf{C} \to \mathbf{C}$ for each $s \in S$.[9]

The λ-complete state operator λ_s allows, given the above-mentioned functions, processes to interact with a state. Let p be a process. Then $\lambda_s(p)$ is the process p executed in state s. The function act gives, for each action a and state s, the action that results from executing a in state s. The function eff gives, for each action a and state s, the state that results from executing a in state s. The function eval gives, for each condition α and state s, the condition that results from evaluating α in state s. The functions act and eff are extended to A_{δ} such that $\mathsf{act}(\delta, s) = \delta$ and $\mathsf{eff}(\delta, s) = s$ for all $s \in S$.

The additional axioms for λ_s, where $s \in S$, are the axioms given in Table 12. Axioms SO1–SO3 are the axioms for the state operators added to ACP in [22].

The structural operational semantics of ACP$^{\mathrm{c}}$ extended with state operators is described by the transition rules for ACP$^{\mathrm{c}}$ and the transition rules given in Table 13.

The full splitting bisimulation models of ACP$^{\mathrm{c}}$ with state operators are simply the expansions of the full splitting bisimulation models $\mathfrak{P}^{\mathrm{c}}_{\kappa}$ of ACP$^{\mathrm{c}}$ obtained by associating with each operator λ_s the corresponding re-labeling operation on conditional transition systems.

[9] Holding on to the usual conventions leads to the double use of the symbol λ: without subscript it stands for an infinite cardinal, and with subscript it stands for a state operator.

Table 12
Axioms for state operators ($a \in \mathsf{A}_\delta$, $\eta \in \mathsf{C}_{\mathsf{at}}$, $\eta' \in \mathsf{C}_{\mathsf{at}} \cup \{\bot, \top\}$)

$\lambda_s(a) = \mathsf{act}(a, s)$	SO1	$\lambda_s(\bot) = \bot$	SO5
$\lambda_s(a \cdot x) = \mathsf{act}(a, s) \cdot \lambda_{\mathsf{eff}(a,s)}(x)$	SO2	$\lambda_s(\top) = \top$	SO6
$\lambda_s(x + y) = \lambda_s(x) + \lambda_s(y)$	SO3	$\lambda_s(\eta) = \eta'$ if $\mathsf{eval}(\eta, s) = \eta'$	SO7
$\lambda_s(\phi :\rightarrow x) = \lambda_s(\phi) :\rightarrow \lambda_s(x)$	SO4	$\lambda_s(-\phi) = -\lambda_s(\phi)$	SO8
		$\lambda_s(\phi \sqcup \psi) = \lambda_s(\phi) \sqcup \lambda_s(\psi)$	SO9
		$\lambda_s(\phi \sqcap \psi) = \lambda_s(\phi) \sqcap \lambda_s(\psi)$	SO10

Table 13
Transition rules for state operators

$$\frac{x \xrightarrow{[\phi]\, a} \surd}{\lambda_s(x) \xrightarrow{[\mathsf{eval}(\phi,s)]\, \mathsf{act}(a,s)} \surd} \quad \mathsf{act}(a, s) \neq \delta,\ \mathsf{eval}(\phi, s) \neq \bot$$

$$\frac{x \xrightarrow{[\phi]\, a} x'}{\lambda_s(x) \xrightarrow{[\mathsf{eval}(\phi,s)]\, \mathsf{act}(a,s)} \lambda_{\mathsf{eff}(a,s)}(x')} \quad \mathsf{act}(a, s) \neq \delta,\ \mathsf{eval}(\phi, s) \neq \bot$$

We can add, in addition to the λ-complete state operators, the λ-complete condition evaluation operators and/or the generalized λ-complete condition evaluation operators from Section 9 to ACP$^{\mathrm{c}}$.

We write $\mathfrak{P}^{\mathrm{ce}\prime}_\kappa$ for the expansion of $\mathfrak{P}^{\mathrm{c}}_\kappa$ for the λ-complete condition evaluation operators, the generalized λ-complete condition evaluation operators and the λ-complete state operators.

The λ-complete state operators are generalizations of the generalized λ-complete condition evaluation operators from Section 9.

Proposition 17 (Generalization) *We can fix S, act, eff and eval such that, for some $f : \mathcal{H}_\lambda \to S$, $\lambda_{f(h)}(x) = \mathsf{GCE}_h(x)$ holds for all $h \in \mathcal{H}_\lambda$ in all full splitting bisimulation models $\mathfrak{P}^{\mathrm{ce}\prime}_\kappa$ with $\kappa \leq \lambda$.*

PROOF. Clearly, if $S = \mathcal{H}_\lambda$, f is the identity function on $\mathcal{H}_\lambda$, and $\mathsf{act}(a, s) = a$, $\mathsf{eff}(a, s) = \mathsf{eff}(a, f^{-1}(s))$ and $\mathsf{eval}(\alpha, s) = f^{-1}(s)(\alpha)$ for all $a \in \mathsf{A}$, $s \in S$ and $\alpha \in \mathcal{C}_\lambda$, then $\lambda_{f(h)}(x) = \mathsf{GCE}_h(x)$ holds for all $h \in \mathcal{H}_\lambda$ in all full splitting bisimulation models $\mathfrak{P}^{\mathrm{ce}\prime}_\kappa$ with $\kappa \leq \lambda$. $\Box$

11 Signal Emission

In Section 9, we made the observation that, if λ is a regular infinite cardinal, condition evaluation by means of the λ-complete condition evaluation operators CE_h from that section is always condition evaluation of which the result can be determined from a set of propositions (see Remark 14). A similar observation can be made about condition evaluation by means of the generalized λ-complete condition evaluation operators GCE_h from that section. In the case of condition evaluation by means of CE_h, the set of propositions determining the result of condition evaluation does not change as a process proceeds. In the case of condition evaluation by means of GCE_h, it may happen that the set of propositions determining the result of condition evaluation changes as a process proceeds. That is, the sets of propositions relevant to a process and its subprocesses may differ. This suggest that condition evaluation can also be dealt with by explicitly associating sets of propositions with processes. The intuition is, then, that all propositions from the set of propositions associated with a process holds at the start of the process.

Clearly, if we restrict ourselves to sets of propositions of cardinality less than a regular infinite cardinal λ, we can associate elements of $\mathcal{C}_\lambda$ with processes instead. In line with [4], the element of $\mathcal{C}_\lambda$ associated with a process is called the signal emitted by the process. Because $\perp$ represents the proposition F, the proposition that cannot hold at the start of any process, we regard a process with which $\perp$ is associated as an inconsistency. However, in an algebraic setting, we cannot exclude this inconsistency. Therefore, we consider it to be a special process, which is called the inaccessible process.[10]

The idea to associate elements of $\mathcal{C}_\lambda$ with processes naturally suggests itself in the case where λ is a regular infinite cardinal. However, there are no trammels to drop the restriction that λ is regular.

All this leads us to an extension of ACP$^{\mathrm{c}}$, called ACP$^{\mathrm{cs}}$, with the following additional constants and operators:

- the *inaccessible process* constant $\perp : \mathbf{P}$;
- the binary *signal emission* operator $\wedge\!\!\!\blacktriangle : \mathbf{C} \times \mathbf{P} \to \mathbf{P}$.

The axioms of ACP$^{\mathrm{cs}}$ are the axioms of ACP$^{\mathrm{c}}$ with axioms CM2–CM3 and GC8–GC10 replaced by axioms CM2S–CM3S and GC8S–GC10S from Table 14, and the additional axioms given in Table 15. Axioms NE1–NE3 and SE1–SE11 have been taken from [6] and axioms GC9S and GC10S have been

[10] In [23,24], this process is rather contradictory called the non-existent process. Its new name was prompted by the fact that after performing an action no process will ever proceed as this process.

Table 14
Axioms adapted to signal emission ($a \in \mathsf{A}_\delta$)

$a \mathbin{\lfloor\!\lfloor} x = a \cdot x + \partial_{\mathsf{A}}(x)$	CM2S
$a \cdot x \mathbin{\lfloor\!\lfloor} y = a \cdot (x \parallel y) + \partial_{\mathsf{A}}(y)$	CM3S
$(\phi :\rightarrow x) \mathbin{\lfloor\!\lfloor} y = \phi :\rightarrow (x \mathbin{\lfloor\!\lfloor} y) + \partial_{\mathsf{A}}(y)$	GC8S
$(\phi :\rightarrow x) \mid y = \phi :\rightarrow (x \mid y) + \partial_{\mathsf{A}}(y)$	GC9S
$x \mid (\phi :\rightarrow y) = \phi :\rightarrow (x \mid y) + \partial_{\mathsf{A}}(x)$	GC10S

Table 15
Additional axioms for signal emission ($a \in \mathsf{A}_\delta$)

$x + \bot = \bot$	NE1	$\phi \mathbin{{}^{\wedge}\!\blacktriangle} (\psi \mathbin{{}^{\wedge}\!\blacktriangle} x) = (\phi \sqcap \psi) \mathbin{{}^{\wedge}\!\blacktriangle} x$	SE5
$\bot \cdot x = \bot$	NE2	$\phi \mathbin{{}^{\wedge}\!\blacktriangle} (\phi :\rightarrow x) = \phi \mathbin{{}^{\wedge}\!\blacktriangle} x$	SE6
$a \cdot \bot = \delta$	NE3	$\phi :\rightarrow (\psi \mathbin{{}^{\wedge}\!\blacktriangle} x) = (-\phi \sqcup \psi) \mathbin{{}^{\wedge}\!\blacktriangle} (\phi :\rightarrow x)$	SE7
$\top \mathbin{{}^{\wedge}\!\blacktriangle} x = x$	SE1	$(\phi \mathbin{{}^{\wedge}\!\blacktriangle} x) \mathbin{\lfloor\!\lfloor} y = \phi \mathbin{{}^{\wedge}\!\blacktriangle} (x \mathbin{\lfloor\!\lfloor} y)$	SE8
$\bot \mathbin{{}^{\wedge}\!\blacktriangle} x = \bot$	SE2	$(\phi \mathbin{{}^{\wedge}\!\blacktriangle} x) \mid y = \phi \mathbin{{}^{\wedge}\!\blacktriangle} (x \mid y)$	SE9
$\phi \mathbin{{}^{\wedge}\!\blacktriangle} x + y = \phi \mathbin{{}^{\wedge}\!\blacktriangle} (x + y)$	SE3	$x \mid (\phi \mathbin{{}^{\wedge}\!\blacktriangle} y) = \phi \mathbin{{}^{\wedge}\!\blacktriangle} (x \mid y)$	SE10
$(\phi \mathbin{{}^{\wedge}\!\blacktriangle} x) \cdot y = \phi \mathbin{{}^{\wedge}\!\blacktriangle} x \cdot y$	SE4	$\partial_H(\phi \mathbin{{}^{\wedge}\!\blacktriangle} x) = \phi \mathbin{{}^{\wedge}\!\blacktriangle} \partial_H(x)$	SE11

taken from [6] with subterms of the form $\mathsf{s}(x) \mathbin{{}^{\wedge}\!\blacktriangle} \delta$ replaced by $\partial_{\mathsf{A}}(x)$. Axioms CM2S, CM3S and GC8S differ really from the corresponding axioms in [6] due to the choice of having as the signal emitted by the left merge of two processes, as in the case of the communication merge, always the meet of the signals emitted by the two processes.

Example 18 *We return to Example 1, which is concerned with a pedestrian who uses a crossing with traffic lights to cross a road with busy traffic safely. In Example 15, we illustrated that the generalized condition evaluation operators* GCE_h *poorly represent the interaction between the pedestrian and the traffic lights. Here, we illustrate that such interaction can be better represented with the signal emission operator* $\mathbin{{}^{\wedge}\!\blacktriangle}$. *Recall that the description of the behaviour of the pedestrian is as follows:*

$$PED = arrive \cdot (green :\rightarrow cross + red :\rightarrow (make\_req \cdot (green :\rightarrow cross)))\ .$$

In the case where the light for pedestrians is red, the traffic lights will grant a request for green light. The next action of the traffic lights is to revert to red light. Suppose that initially the light for pedestrians is red. Then the behaviour of the traffic lights can be described as follows:

$$TL = (red \sqcap -green) \mathbin{{}^{\wedge}\!\blacktriangle} grant\_req \cdot ((-red \sqcap green) \mathbin{{}^{\wedge}\!\blacktriangle} revert \cdot TL)\ .$$

Let the communication function $\mid$ *be such that* $\mathit{make\_req} \mid \mathit{grant\_req} = \mathit{grant\_req} \mid \mathit{make\_req} = \mathit{request}$ *and* $a \mid b = \delta$ *otherwise. Then we can derive the following about the combined behaviour of the pedestrian and the traffic lights:*

$$
\begin{aligned}
&\partial_{\{\mathit{make\_req},\mathit{grant\_req}\}}(\mathit{PED} \parallel \mathit{TL}) = \\
&\quad (\mathit{red} \sqcap -\mathit{green}) \wedge \mathit{arrive} \cdot ((\mathit{red} \sqcap -\mathit{green}) \wedge \mathit{request} \cdot \\
&\quad\quad ((-\mathit{red} \sqcap \mathit{green}) \wedge \mathit{cross} \cdot ((-\mathit{red} \sqcap \mathit{green}) \wedge \mathit{revert} \cdot \mathit{TL}))) + \\
&\quad (\mathit{red} \sqcap -\mathit{green}) \wedge \mathit{arrive} \cdot ((\mathit{red} \sqcap -\mathit{green}) \wedge \mathit{request} \cdot \\
&\quad\quad ((-\mathit{red} \sqcap \mathit{green}) \wedge \mathit{revert} \cdot ((\mathit{red} \sqcap -\mathit{green}) \wedge \delta)))\ .
\end{aligned}
$$

The possibility that the combined behaviour ends in deadlock shows that we have actually described a rather simple-minded pedestrian. If he or she has not started crossing the road before the traffic lights revert to red light, the pedestrian takes no action; and consequently the light remains red. This remains unobserved in the case where the interaction between the pedestrian and the traffic lights is represented by the generalized condition evaluation operators GCE_h*.*

In the structural operational semantics of $\mathrm{ACP}^{\mathrm{cs}}$, unary relations s^α, one for each $\alpha \in \mathcal{C} \setminus \{\bot\}$, are used in addition to the relations $\xrightarrow{\ell}\surd$ and $\xrightarrow{\ell}$. We write $\mathsf{s}(p) = \alpha$ instead of $p \in \mathsf{s}^\alpha$. The relation s^α can be explained as follows:

- $\mathsf{s}(p) = \alpha$: p emits the signal α.

The structural operational semantics of $\mathrm{ACP}^{\mathrm{cs}}$ is described by the transition rules given in Tables 16 and 17. These transition rules include all transition rules from Tables 3 and 5 with additional premises to exclude transitions from or to processes that emit the signal $\bot$. There are additional transition rules describing the signals emitted by the processes. The transition rules for signal emission are new as well.

The following gives a good picture of the nature of signals and conditions.

Proposition 19 (Signals and conditions) *If* $\langle\!\langle \alpha \rangle\!\rangle \vdash \langle\!\langle \beta \rangle\!\rangle \Leftrightarrow \langle\!\langle \beta' \rangle\!\rangle$*, then* $\underline{\alpha} \wedge (\underline{\beta} :\rightarrow x) = \underline{\alpha} \wedge (\underline{\beta'} :\rightarrow x)$.

PROOF. It follows immediately from $\langle\!\langle \alpha \rangle\!\rangle \vdash \langle\!\langle \beta \rangle\!\rangle \Leftrightarrow \langle\!\langle \beta' \rangle\!\rangle$, using the deduction theorem of propositional calculus and the isomorphism of $\mathcal{C}$ and the Boolean algebra of equivalence classes with respect to logical equivalence of the set of all finite propositions with elements of $\mathsf{C_{at}}$ as propositional variables, that $(-(\alpha \sqcap \beta) \sqcup \beta') \sqcap (-(\alpha \sqcap \beta') \sqcup \beta) = \top$. It follows easily from this equation, using the axioms of BA, that $(\underline{\alpha} \sqcap \underline{\beta}) \sqcup \underline{\beta'} = \underline{\beta'}$ (*) and

Table 16
Transition rules for $\mathrm{BPA}_{\delta}^{\mathrm{cs}}$

$$\frac{}{a \xrightarrow{[\top]\, a} \surd}$$

$$\frac{x \xrightarrow{[\phi]\, a} \surd,\ \mathsf{s}(x+y) \neq \bot}{x+y \xrightarrow{[\phi]\, a} \surd} \qquad \frac{y \xrightarrow{[\phi]\, a} \surd,\ \mathsf{s}(x+y) \neq \bot}{x+y \xrightarrow{[\phi]\, a} \surd}$$

$$\frac{x \xrightarrow{[\phi]\, a} x',\ \mathsf{s}(x+y) \neq \bot}{x+y \xrightarrow{[\phi]\, a} x'} \qquad \frac{y \xrightarrow{[\phi]\, a} y',\ \mathsf{s}(x+y) \neq \bot}{x+y \xrightarrow{[\phi]\, a} y'}$$

$$\frac{x \xrightarrow{[\phi]\, a} \surd,\ \mathsf{s}(y) \neq \bot}{x \cdot y \xrightarrow{[\phi]\, a} y} \qquad \frac{x \xrightarrow{[\phi]\, a} x'}{x \cdot y \xrightarrow{[\phi]\, a} x' \cdot y}$$

$$\frac{x \xrightarrow{[\phi]\, a} \surd}{\psi :\to x \xrightarrow{[\phi \sqcap \psi]\, a} \surd}\ \phi \sqcap \psi \neq \bot \qquad \frac{x \xrightarrow{[\phi]\, a} x'}{\psi :\to x \xrightarrow{[\phi \sqcap \psi]\, a} x'}\ \phi \sqcap \psi \neq \bot$$

$$\frac{x \xrightarrow{[\phi]\, a} \surd,\ \mathsf{s}(\psi \mathbin{{}^{\wedge}\!\!\blacktriangle} x) \neq \bot}{\psi \mathbin{{}^{\wedge}\!\!\blacktriangle} x \xrightarrow{[\phi]\, a} \surd} \qquad \frac{x \xrightarrow{[\phi]\, a} x',\ \mathsf{s}(\psi \mathbin{{}^{\wedge}\!\!\blacktriangle} x) \neq \bot}{\psi \mathbin{{}^{\wedge}\!\!\blacktriangle} x \xrightarrow{[\phi]\, a} x'}$$

$$\frac{}{\mathsf{s}(\bot) = \bot} \qquad \frac{}{\mathsf{s}(a) = \top} \qquad \frac{\mathsf{s}(x) = \phi,\ \mathsf{s}(y) = \psi}{\mathsf{s}(x+y) = \phi \sqcap \psi} \qquad \frac{\mathsf{s}(x) = \phi}{\mathsf{s}(x \cdot y) = \phi}$$

$$\frac{\mathsf{s}(x) = \phi}{\mathsf{s}(\psi :\to y) = -\psi \sqcup \phi} \qquad \frac{\mathsf{s}(x) = \phi}{\mathsf{s}(\psi \mathbin{{}^{\wedge}\!\!\blacktriangle} y) = \psi \sqcap \phi}$$

$(\underline{\alpha} \sqcap \underline{\beta'}) \sqcup \underline{\beta} = \underline{\beta}$ (**). From (*), using the axioms of $\mathrm{ACP}^{\mathrm{cs}}$, we can derive $\underline{\alpha} \mathbin{{}^{\wedge}\!\!\blacktriangle} (\underline{\beta} :\to x) + \underline{\alpha} \mathbin{{}^{\wedge}\!\!\blacktriangle} (\underline{\beta'} :\to x) = \underline{\alpha} \mathbin{{}^{\wedge}\!\!\blacktriangle} (\underline{\beta'} :\to x)$ as follows:

$$\begin{aligned}
&\underline{\alpha} \mathbin{{}^{\wedge}\!\!\blacktriangle} (\underline{\beta} :\to x) + \underline{\alpha} \mathbin{{}^{\wedge}\!\!\blacktriangle} (\underline{\beta'} :\to x) \\
&\quad \overset{\text{SE6,GC6}}{=} \quad \underline{\alpha} \mathbin{{}^{\wedge}\!\!\blacktriangle} (\underline{\alpha} \sqcap \underline{\beta} :\to x) + \underline{\alpha} \mathbin{{}^{\wedge}\!\!\blacktriangle} (\underline{\beta'} :\to x) \\
&\quad \overset{\text{SE3,SE5,BA,GC7}}{=} \quad \underline{\alpha} \mathbin{{}^{\wedge}\!\!\blacktriangle} ((\underline{\alpha} \sqcap \underline{\beta}) \sqcup \underline{\beta'} :\to x) \\
&\quad \overset{(*)}{=} \quad \underline{\alpha} \mathbin{{}^{\wedge}\!\!\blacktriangle} (\underline{\beta'} :\to x)\ .
\end{aligned}$$

From (**), we can derive analogously $\underline{\alpha} \mathbin{{}^{\wedge}\!\!\blacktriangle} (\underline{\beta'} :\to x) + \underline{\alpha} \mathbin{{}^{\wedge}\!\!\blacktriangle} (\underline{\beta} :\to x) = \underline{\alpha} \mathbin{{}^{\wedge}\!\!\blacktriangle} (\underline{\beta} :\to x)$. From these two results, it follows immediately that $\underline{\alpha} \mathbin{{}^{\wedge}\!\!\blacktriangle} (\underline{\beta} :\to x) = \underline{\alpha} \mathbin{{}^{\wedge}\!\!\blacktriangle} (\underline{\beta'} :\to x)$. □

We have the following corollaries from Proposition 19.

Corollary 20 *If* $\langle\!\langle \alpha \rangle\!\rangle \vdash \langle\!\langle \beta \rangle\!\rangle$, *then* $\underline{\alpha} \mathbin{{}^{\wedge}\!\!\blacktriangle} (\underline{\beta} :\to x) = \underline{\alpha} \mathbin{{}^{\wedge}\!\!\blacktriangle} x$. *If* $\langle\!\langle \alpha \rangle\!\rangle \vdash \neg \langle\!\langle \beta \rangle\!\rangle$, *then* $\underline{\alpha} \mathbin{{}^{\wedge}\!\!\blacktriangle} (\underline{\beta} :\to x) = \underline{\alpha} \mathbin{{}^{\wedge}\!\!\blacktriangle} \delta$.

Table 17
Additional transition rules for ACP$^{\mathrm{cs}}$

$$\frac{x \xrightarrow{[\phi]\,a} \surd,\ \mathsf{s}(x \parallel y) \neq \bot,\ \mathsf{s}(y) \neq \bot}{x \parallel y \xrightarrow{[\phi]\,a} y} \qquad \frac{y \xrightarrow{[\phi]\,a} \surd,\ \mathsf{s}(x \parallel y) \neq \bot,\ \mathsf{s}(x) \neq \bot}{x \parallel y \xrightarrow{[\phi]\,a} x}$$

$$\frac{x \xrightarrow{[\phi]\,a} x',\ \mathsf{s}(x \parallel y) \neq \bot,\ \mathsf{s}(x' \parallel y) \neq \bot}{x \parallel y \xrightarrow{[\phi]\,a} x' \parallel y} \qquad \frac{y \xrightarrow{[\phi]\,a} y',\ \mathsf{s}(x \parallel y) \neq \bot,\ \mathsf{s}(x \parallel y') \neq \bot}{x \parallel y \xrightarrow{[\phi]\,a} x \parallel y'}$$

$$\frac{x \xrightarrow{[\phi]\,a} \surd,\ y \xrightarrow{[\psi]\,b} \surd,\ \mathsf{s}(x \parallel y) \neq \bot}{x \parallel y \xrightarrow{[\phi \sqcap \psi]\,c} \surd}\ a \mid b = c,\ \phi \sqcap \psi \neq \bot$$

$$\frac{x \xrightarrow{[\phi]\,a} \surd,\ y \xrightarrow{[\psi]\,b} y',\ \mathsf{s}(x \parallel y) \neq \bot}{x \parallel y \xrightarrow{[\phi \sqcap \psi]\,c} y'}\ a \mid b = c,\ \phi \sqcap \psi \neq \bot$$

$$\frac{x \xrightarrow{[\phi]\,a} x',\ y \xrightarrow{[\psi]\,b} \surd,\ \mathsf{s}(x \parallel y) \neq \bot}{x \parallel y \xrightarrow{[\phi \sqcap \psi]\,c} x'}\ a \mid b = c,\ \phi \sqcap \psi \neq \bot$$

$$\frac{x \xrightarrow{[\phi]\,a} x',\ y \xrightarrow{[\psi]\,b} y',\ \mathsf{s}(x \parallel y) \neq \bot,\ \mathsf{s}(x' \parallel y') \neq \bot}{x \parallel y \xrightarrow{[\phi \sqcap \psi]\,c} x' \parallel y'}\ a \mid b = c,\ \phi \sqcap \psi \neq \bot$$

$$\frac{x \xrightarrow{[\phi]\,a} \surd,\ \mathsf{s}(x \mathbin{\lfloor\!\lfloor} y) \neq \bot,\ \mathsf{s}(y) \neq \bot}{x \mathbin{\lfloor\!\lfloor} y \xrightarrow{[\phi]\,a} y} \qquad \frac{x \xrightarrow{[\phi]\,a} x',\ \mathsf{s}(x \mathbin{\lfloor\!\lfloor} y) \neq \bot,\ \mathsf{s}(x' \parallel y) \neq \bot}{x \mathbin{\lfloor\!\lfloor} y \xrightarrow{[\phi]\,a} x' \parallel y}$$

$$\frac{x \xrightarrow{[\phi]\,a} \surd,\ y \xrightarrow{[\psi]\,b} \surd,\ \mathsf{s}(x \mid y) \neq \bot}{x \mid y \xrightarrow{[\phi \sqcap \psi]\,c} \surd}\ a \mid b = c,\ \phi \sqcap \psi \neq \bot$$

$$\frac{x \xrightarrow{[\phi]\,a} \surd,\ y \xrightarrow{[\psi]\,b} y',\ \mathsf{s}(x \mid y) \neq \bot}{x \mid y \xrightarrow{[\phi \sqcap \psi]\,c} y'}\ a \mid b = c,\ \phi \sqcap \psi \neq \bot$$

$$\frac{x \xrightarrow{[\phi]\,a} x',\ y \xrightarrow{[\psi]\,b} \surd,\ \mathsf{s}(x \mid y) \neq \bot}{x \mid y \xrightarrow{[\phi \sqcap \psi]\,c} x'}\ a \mid b = c,\ \phi \sqcap \psi \neq \bot$$

$$\frac{x \xrightarrow{[\phi]\,a} x',\ y \xrightarrow{[\psi]\,b} y',\ \mathsf{s}(x \mid y) \neq \bot,\ \mathsf{s}(x' \parallel y') \neq \bot}{x \mid y \xrightarrow{[\phi \sqcap \psi]\,c} x' \parallel y'}\ a \mid b = c,\ \phi \sqcap \psi \neq \bot$$

$$\frac{x \xrightarrow{[\phi]\,a} \surd}{\partial_H(x) \xrightarrow{[\phi]\,a} \surd}\ a \notin H \qquad \frac{x \xrightarrow{[\phi]\,a} x'}{\partial_H(x) \xrightarrow{[\phi]\,a} \partial_H(x')}\ a \notin H$$

$$\frac{\mathsf{s}(x) = \phi,\ \mathsf{s}(y) = \psi}{\mathsf{s}(x \parallel y) = \phi \sqcap \psi} \qquad \frac{\mathsf{s}(x) = \phi,\ \mathsf{s}(y) = \psi}{\mathsf{s}(x \mathbin{\lfloor\!\lfloor} y) = \phi \sqcap \psi} \qquad \frac{\mathsf{s}(x) = \phi,\ \mathsf{s}(y) = \psi}{\mathsf{s}(x \mid y) = \phi \sqcap \psi} \qquad \frac{\mathsf{s}(x) = \phi}{\mathsf{s}(\partial_H(x)) = \phi}$$

Corollary 21 *If* $\mathsf{eff}(h, a)$ *is the identity endomorphism on* $\mathcal{C}$ *for all endomorphisms* h *on* $\mathcal{C}$ *and* $a \in \mathsf{A}$*, then we have* $\mathsf{GCE}_{h_{\{\langle\!\langle \alpha \rangle\!\rangle\}}}(\underline{\beta} {:}\!\rightarrow x) = \underline{\beta'} {:}\!\rightarrow \mathsf{GCE}_{h_{\{\langle\!\langle \alpha \rangle\!\rangle\}}}(x)$ *implies* $\underline{\alpha} \mathbin{{}^{\wedge\!\blacktriangle}} (\underline{\beta} {:}\!\rightarrow x) = \underline{\alpha} \mathbin{{}^{\wedge\!\blacktriangle}} (\underline{\beta'} {:}\!\rightarrow x)$*.*

12 Full Signal-Observing Splitting Bisimulation Models of ACP$^{\mathrm{cs}}$

In this section, we introduce conditional transition systems with signals, signal-observing splitting bisimilarity of conditional transition systems with signals, and the full signal-observing splitting bisimulation models of ACP$^{\mathrm{cs}}$.

Conditional transition systems with signals generalize conditional transition systems.

Let κ be an infinite cardinal. Then a *κ-conditional transition system with signals T* is a tuple $(S, \rightarrow, \rightarrow\surd, \mathsf{s}, s^0)$ where

- $(S, \rightarrow, \rightarrow\surd, s^0)$ is a κ-conditional transition system;
- s is a function from S to $\mathcal{C}_\kappa$;

and for all $\ell \in \mathcal{C}_\kappa^- \times \mathsf{A}$:

- $\{(s, s') \in \xrightarrow{\ell} \mid \mathsf{s}(s) = \bot \lor \mathsf{s}(s') = \bot\} = \emptyset$;
- $\{s \in \xrightarrow{\ell}\surd \mid \mathsf{s}(s) = \bot\} = \emptyset$.

We say that $\mathsf{s}(s)$ is the signal emitted by the state s.

For conditional transition systems with signals, reachability and connectedness are defined exactly as for conditional transition systems.

Let $T = (S, \rightarrow, \rightarrow\surd, \mathsf{s}, s^0)$ be a κ-conditional transition system with signals (for an infinite cardinal κ) that is not necessarily connected. Then the *connected part* of T, written $\Gamma(T)$, is simply defined as follows:

$$\Gamma(T) = (S', \rightarrow', \rightarrow\surd', \mathsf{s}', s^0) \;,$$

where

$$(S', \rightarrow', \rightarrow\surd', s^0) = \Gamma(S, \rightarrow, \rightarrow\surd, s^0) \;,$$
$$\mathsf{s}' \text{ is the restriction of } \mathsf{s} \text{ to } S' \;.$$

Let κ be an infinite cardinal. Then $\mathbb{CTS}_\kappa^{\mathrm{s}}$ is the set of all κ-conditional transition systems with signals $(S, \rightarrow, \rightarrow\surd, \mathsf{s}, s^0)$ for which $(S, \rightarrow, \rightarrow\surd, s^0) \in \mathbb{CTS}_\kappa$.

Isomorphism between conditional transition systems with signals is defined as between conditional transition systems, but with the additional condition that $\mathsf{s}_1(s) = \mathsf{s}_2(b(s))$. Splitting bisimilarity has to be adapted to the setting with signals.

Let $T_1 = (S_1, \rightarrow_1, \rightarrow\surd_1, \mathsf{s}_1, s_1^0) \in \mathbb{CTS}^{\mathrm{s}}_\kappa$, $T_2 = (S_2, \rightarrow_2, \rightarrow\surd_2, \mathsf{s}_2, s_2^0) \in \mathbb{CTS}^{\mathrm{s}}_\kappa$ (for an infinite cardinal κ). Then a *signal-observing splitting bisimulation* B between T_1 and T_2 is a binary relation $B \subseteq S_1 \times S_2$ such that $B(s_1^0, s_2^0)$ and for all s_1, s_2 such that $B(s_1, s_2)$:

- $\mathsf{s}_1(s_1) = \mathsf{s}_2(s_2)$;
- if $s_1 \xrightarrow{[\alpha]\,a}_1 s_1'$, then there is a set $CS_2' \subseteq \mathcal{C}_\kappa^- \times S_2$ of cardinality less than κ such that $\mathsf{s}_1(s_1) \sqcap \alpha \sqsubseteq \bigsqcup \mathrm{dom}(CS_2')$ and for all $(\alpha', s_2') \in CS_2'$, $s_2 \xrightarrow{[\alpha']\,a}_2 s_2'$ and $B(s_1', s_2')$;
- if $s_2 \xrightarrow{[\alpha]\,a}_2 s_2'$, then there is a set $CS_1' \subseteq \mathcal{C}_\kappa^- \times S_1$ of cardinality less than κ such that $\mathsf{s}_2(s_2) \sqcap \alpha \sqsubseteq \bigsqcup \mathrm{dom}(CS_1')$ and for all $(\alpha', s_1') \in CS_1'$, $s_1 \xrightarrow{[\alpha']\,a}_1 s_1'$ and $B(s_1', s_2')$;
- if $s_1 \xrightarrow{[\alpha]\,a} \surd_1$, then there is a set $C' \subseteq \mathcal{C}_\kappa^-$ of cardinality less than κ such that $\mathsf{s}_1(s_1) \sqcap \alpha \sqsubseteq \bigsqcup C'$ and for all $\alpha' \in C'$, $s_2 \xrightarrow{[\alpha']\,a} \surd_2$;
- if $s_2 \xrightarrow{[\alpha]\,a} \surd_2$, then there is a set $C' \subseteq \mathcal{C}_\kappa^-$ of cardinality less than κ such that $\mathsf{s}_2(s_2) \sqcap \alpha \sqsubseteq \bigsqcup C'$ and for all $\alpha' \in C'$, $s_1 \xrightarrow{[\alpha']\,a} \surd_1$.

Two conditional transition systems with signals $T_1, T_2 \in \mathbb{CTS}^{\mathrm{s}}_\kappa$ are *signal-observing splitting bisimilar*, written $T_1 \leftrightarrow\!\!\!\!\!\!\!\;-^{\mathrm{s}} T_2$, if there exists a signal-observing splitting bisimulation B between T_1 and T_2. Let B be a signal-observing splitting bisimulation between T_1 and T_2. Then we say that B is a splitting signal-observing bisimulation *witnessing* $T_1 \leftrightarrow\!\!\!\!\!\!\!\;-^{\mathrm{s}} T_2$.

It is straightforward to see that $\leftrightarrow\!\!\!\!\!\!\!\;-^{\mathrm{s}}$ is an equivalence on $\mathbb{CTS}^{\mathrm{s}}_\kappa$. Let $T \in \mathbb{CTS}^{\mathrm{s}}_\kappa$. Then we write $[T]_{\leftrightarrow^{\mathrm{s}}}$ for $\{T' \in \mathbb{CTS}^{\mathrm{s}}_\kappa \mid T \leftrightarrow\!\!\!\!\!\!\!\;-^{\mathrm{s}} T'\}$, i.e. the $\leftrightarrow\!\!\!\!\!\!\!\;-^{\mathrm{s}}$-equivalence class of T. We write $\mathbb{CTS}^{\mathrm{s}}_\kappa/\leftrightarrow\!\!\!\!\!\!\!\;-^{\mathrm{s}}$ for the set of equivalence classes $\{[T]_{\leftrightarrow^{\mathrm{s}}} \mid T \in \mathbb{CTS}^{\mathrm{s}}_\kappa\}$.

The elements of $\mathbb{CTS}^{\mathrm{s}}_\kappa$ and operations on $\mathbb{CTS}^{\mathrm{s}}_\kappa$ to be associated with the constants δ and a (for each $a \in \mathsf{A}$) and the operators $+$, $\cdot$, $:\rightarrow$, $\|$, $\lfloor\!\lfloor$, $\mid$ and ∂_H (for each $H \subseteq \mathsf{A}$) are as the elements of $\mathbb{CTS}_\kappa$ and operations on $\mathbb{CTS}_\kappa$ associated with them before, but with the additional function s as suggested by the structural operational semantics of ACP$^{\mathrm{cs}}$ and with all relations $\xrightarrow{\ell}\surd$ and $\xrightarrow{\ell}$ restricted to states that emit a signal different from $\bot$.

We associate with the additional constant $\bot$ an element $\widehat{\bot}^{\mathrm{s}}$ of $\mathbb{CTS}^{\mathrm{s}}_\kappa$ and with the additional operator $\curlywedge$ an operation $\widehat{\curlywedge}^{\mathrm{s}}$ on $\mathbb{CTS}^{\mathrm{s}}_\kappa$ as follows.

- $$\widehat{\bot}^{\mathrm{s}} = (\{s^0\}, \emptyset, \emptyset, \mathsf{s}, s^0)\;,$$
  where
  $$\mathsf{s}(s^0) = \bot\;.$$

- Let $\alpha \in \mathcal{C}$ and $T = (S, \rightarrow, \rightarrow\surd, \mathsf{s}, s^0) \in \mathbb{CTS}^{\mathrm{s}}_{\kappa}$. Then

  $$\alpha \mathbin{\widehat{\overset{\wedge}{\blacktriangle}}^{\mathrm{s}}} T = \Gamma(S, \rightarrow', \rightarrow\surd', \mathsf{s}', s^0) \;,$$

  where

  $$\begin{aligned} \mathsf{s}'(s) &= \mathsf{s}(s) && \text{for } s \in S \setminus \{s^0\} \;, \\ \mathsf{s}'(s^0) &= \alpha \sqcap \mathsf{s}(s^0) \;, \end{aligned}$$

  and for every $(\alpha', a) \in \mathcal{C}^{-}_{\kappa} \times \mathsf{A}$:

  $$\begin{aligned} \xrightarrow{(\alpha',a)}{}' &= \Big\{(s, s') \;\Big|\; s \xrightarrow{[\alpha']\,a} s' \wedge \mathsf{s}'(s) \neq \bot \wedge \mathsf{s}'(s') \neq \bot\Big\} \;, \\ \xrightarrow{(\alpha',a)}\surd' &= \Big\{s \;\Big|\; s \xrightarrow{[\alpha']\,a}\surd \wedge \mathsf{s}'(s) \neq \bot\Big\} \;. \end{aligned}$$

We can easily show that signal-observing splitting bisimilarity is a congruence with respect to alternative composition, sequential composition, guarded command, signal emission, parallel composition, left merge, communication merge and encapsulation.

Proposition 22 (Congruence) *Let κ be an infinite cardinal. Then for all $T_1, T_2, T_1', T_2' \in \mathbb{CTS}_{\kappa}$ and $\alpha \in \mathcal{C}$, $T_1 \mathrel{\underline{\leftrightarrow}}^{\mathrm{s}} T_1'$ and $T_2 \mathrel{\underline{\leftrightarrow}}^{\mathrm{s}} T_2'$ imply $T_1 \mathbin{\widehat{+}^{\mathrm{s}}} T_2 \mathrel{\underline{\leftrightarrow}}^{\mathrm{s}} T_1' \mathbin{\widehat{+}^{\mathrm{s}}} T_2'$, $T_1 \mathbin{\widehat{\cdot}^{\mathrm{s}}} T_2 \mathrel{\underline{\leftrightarrow}}^{\mathrm{s}} T_1' \mathbin{\widehat{\cdot}^{\mathrm{s}}} T_2'$, $\alpha \mathbin{\widehat{:\rightarrow}^{\mathrm{s}}} T_1 \mathrel{\underline{\leftrightarrow}}^{\mathrm{s}} \alpha \mathbin{\widehat{:\rightarrow}^{\mathrm{s}}} T_1'$, $\alpha \mathbin{\widehat{\overset{\wedge}{\blacktriangle}}^{\mathrm{s}}} T_1 \mathrel{\underline{\leftrightarrow}}^{\mathrm{s}} \alpha \mathbin{\widehat{\overset{\wedge}{\blacktriangle}}^{\mathrm{s}}} T_1'$, $T_1 \mathbin{\widehat{\parallel}^{\mathrm{s}}} T_2 \mathrel{\underline{\leftrightarrow}}^{\mathrm{s}} T_1' \mathbin{\widehat{\parallel}^{\mathrm{s}}} T_2'$, $T_1 \mathbin{\widehat{\lfloor\!\lfloor}^{\mathrm{s}}} T_2 \mathrel{\underline{\leftrightarrow}}^{\mathrm{s}} T_1' \mathbin{\widehat{\lfloor\!\lfloor}^{\mathrm{s}}} T_2'$, $T_1 \mathbin{\widehat{\mid}^{\mathrm{s}}} T_2 \mathrel{\underline{\leftrightarrow}}^{\mathrm{s}} T_1' \mathbin{\widehat{\mid}^{\mathrm{s}}} T_2'$ and $\widehat{\partial_H}^{\mathrm{s}}(T_1) \mathrel{\underline{\leftrightarrow}}^{\mathrm{s}} \widehat{\partial_H}^{\mathrm{s}}(T_1')$.*

PROOF. For $\widehat{+}^{\mathrm{s}}$, $\widehat{\cdot}^{\mathrm{s}}$ and $\widehat{:\rightarrow}^{\mathrm{s}}$, witnessing signal-observing splitting bisimulations are constructed in the same way as witnessing splitting bisimulations are constructed in the proof of Proposition 5. For $\widehat{\parallel}^{\mathrm{s}}$, $\widehat{\lfloor\!\lfloor}^{\mathrm{s}}$, $\widehat{\mid}^{\mathrm{s}}$ and $\widehat{\partial_H}^{\mathrm{s}}$, witnessing signal-observing splitting bisimulations are constructed in the same way as witnessing splitting bisimulations are constructed in the proof of Proposition 8. What remains is to construct a witnessing signal-observing splitting bisimulation for $\widehat{\overset{\wedge}{\blacktriangle}}^{\mathrm{s}}$. That is simple. Let R be a signal-observing splitting bisimulation witnessing $T_1 \mathrel{\underline{\leftrightarrow}}^{\mathrm{s}} T_1'$. Then we construct a relation $R_{\widehat{\overset{\wedge}{\blacktriangle}}^{\mathrm{s}}}$ as follows:

- $R_{\widehat{\overset{\wedge}{\blacktriangle}}^{\mathrm{s}}} = R \cap (S \times S')$, where S and S' are the sets of states of $\alpha \mathbin{\widehat{\overset{\wedge}{\blacktriangle}}^{\mathrm{s}}} T_1$ and $\alpha \mathbin{\widehat{\overset{\wedge}{\blacktriangle}}^{\mathrm{s}}} T_1'$, respectively.

Given the definition of signal emission, it is easy to see that $R_{\widehat{\overset{\wedge}{\blacktriangle}}^{\mathrm{s}}}$ is a signal-observing splitting bisimulation witnessing $\alpha \mathbin{\widehat{\overset{\wedge}{\blacktriangle}}^{\mathrm{s}}} T_1 \mathrel{\underline{\leftrightarrow}}^{\mathrm{s}} \alpha \mathbin{\widehat{\overset{\wedge}{\blacktriangle}}^{\mathrm{s}}} T_1'$. $\square$

The *full signal-observing splitting bisimulation models* $\mathfrak{P}^{\mathrm{cs}}_{\kappa}$ of ACP$^{\mathrm{cs}}$, one for each infinite cardinal κ, are the expansions of $\mathcal{C}$ whose additional ingredients are defined as follows:

$$
\begin{array}{llll}
\mathcal{P} & = \mathbb{CTS}^{\mathrm{s}}_{\kappa}/\underline{\leftrightarrow}^{\mathrm{s}}\ , & \alpha \mathbin{\widetilde{:\rightarrow}}^{\mathrm{s}} [\,T_1\,]_{\underline{\leftrightarrow}^{\mathrm{s}}} & = [\,\alpha \mathbin{\widehat{:\rightarrow}}^{\mathrm{s}} T_1\,]_{\underline{\leftrightarrow}^{\mathrm{s}}}\ , \\
\widetilde{\bot}^{\mathrm{s}} & = [\,\widehat{\bot}^{\mathrm{s}}\,]_{\underline{\leftrightarrow}^{\mathrm{s}}}\ , & \alpha \mathbin{\widetilde{\wedge\!\blacktriangle}}^{\mathrm{s}} [\,T_1\,]_{\underline{\leftrightarrow}^{\mathrm{s}}} & = [\,\alpha \mathbin{\widehat{\wedge\!\blacktriangle}}^{\mathrm{s}} T_1\,]_{\underline{\leftrightarrow}^{\mathrm{s}}}\ , \\
\widetilde{\delta}^{\mathrm{s}} & = [\,\widehat{\delta}^{\mathrm{s}}\,]_{\underline{\leftrightarrow}^{\mathrm{s}}}\ , & [\,T_1\,]_{\underline{\leftrightarrow}^{\mathrm{s}}} \mathbin{\widetilde{\|}}^{\mathrm{s}} [\,T_2\,]_{\underline{\leftrightarrow}^{\mathrm{s}}} & = [\,T_1 \mathbin{\widehat{\|}}^{\mathrm{s}} T_2\,]_{\underline{\leftrightarrow}^{\mathrm{s}}}\ , \\
\widetilde{a}^{\mathrm{s}} & = [\,\widehat{a}^{\mathrm{s}}\,]_{\underline{\leftrightarrow}^{\mathrm{s}}}\ , & [\,T_1\,]_{\underline{\leftrightarrow}^{\mathrm{s}}} \mathbin{\widetilde{\|\!\lfloor}}^{\mathrm{s}} [\,T_2\,]_{\underline{\leftrightarrow}^{\mathrm{s}}} & = [\,T_1 \mathbin{\widehat{\|\!\lfloor}}^{\mathrm{s}} T_2\,]_{\underline{\leftrightarrow}^{\mathrm{s}}}\ , \\
[\,T_1\,]_{\underline{\leftrightarrow}^{\mathrm{s}}} \mathbin{\widetilde{+}}^{\mathrm{s}} [\,T_2\,]_{\underline{\leftrightarrow}^{\mathrm{s}}} & = [\,T_1 \mathbin{\widehat{+}}^{\mathrm{s}} T_2\,]_{\underline{\leftrightarrow}^{\mathrm{s}}}\ , & [\,T_1\,]_{\underline{\leftrightarrow}^{\mathrm{s}}} \mathbin{\widetilde{\mid}}^{\mathrm{s}} [\,T_2\,]_{\underline{\leftrightarrow}^{\mathrm{s}}} & = [\,T_1 \mathbin{\widehat{\mid}}^{\mathrm{s}} T_2\,]_{\underline{\leftrightarrow}^{\mathrm{s}}}\ , \\
[\,T_1\,]_{\underline{\leftrightarrow}^{\mathrm{s}}} \mathbin{\widetilde{\cdot}}^{\mathrm{s}} [\,T_2\,]_{\underline{\leftrightarrow}^{\mathrm{s}}} & = [\,T_1 \mathbin{\widehat{\cdot}}^{\mathrm{s}} T_2\,]_{\underline{\leftrightarrow}^{\mathrm{s}}}\ , & \widetilde{\partial_H}^{\mathrm{s}}([\,T_1\,]_{\underline{\leftrightarrow}^{\mathrm{s}}}) & = [\,\widehat{\partial_H}^{\mathrm{s}}(T_1)\,]_{\underline{\leftrightarrow}^{\mathrm{s}}}\ .
\end{array}
$$

Alternative composition, sequential composition, guarded command, signal emission, parallel composition, left merge, communication merge and encapsulation on $\mathbb{CTS}^{\mathrm{s}}_{\kappa}/\underline{\leftrightarrow}^{\mathrm{s}}$ are well-defined because $\underline{\leftrightarrow}^{\mathrm{s}}$ is a congruence with respect to the corresponding operations on $\mathbb{CTS}^{\mathrm{s}}_{\kappa}$.

The structures $\mathfrak{P}^{\mathrm{cs}}_{\kappa}$ are models of ACP$^{\mathrm{cs}}$.

Theorem 23 (Soundness of ACP$^{\mathrm{cs}}$) *For each infinite cardinal κ, we have* $\mathfrak{P}^{\mathrm{cs}}_{\kappa} \models \mathrm{ACP}^{\mathrm{cs}}$.

PROOF. Because $\mathfrak{P}^{\mathrm{cs}}_{\kappa}$ is an expansion of $\mathcal{C}$, it is not necessary to show that the axioms of BA are sound. The soundness of all remaining axioms follows straightforwardly from the definitions of the ingredients of $\mathfrak{P}^{\mathrm{cs}}_{\kappa}$. $\square$

13 BPA with Retrospective Conditions

In this section, we present an extension of BPA$^{\mathrm{c}}_{\delta}$ with a retrospection operator on conditions. The retrospection operator allows for looking back on conditions under which preceding actions have been performed. The extension of BPA$^{\mathrm{c}}_{\delta}$ with the retrospection operator is called BPA$^{\mathrm{cr}}_{\delta}$. In Section 15, we will add parallel composition and encapsulation to BPA$^{\mathrm{cr}}_{\delta}$.

BPA$^{\mathrm{cr}}_{\delta}$ has the constants and operators of BPA$^{\mathrm{c}}_{\delta}$ and in addition:

- the unary *retrospection* operator $\sim : \mathbf{C} \to \mathbf{C}$.

Table 18
Additional axioms for retrospection operator ($a \in \mathsf{A}_\delta$)

$\sim\!\bot = \bot$	R1
$\sim\!\top = \top$	R2
$\sim\!(-\phi) = -(\sim\!\phi)$	R3
$\sim\!(\phi \sqcup \psi) = \sim\!\phi \sqcup \sim\!\psi$	R4
$\sim\!(\phi \sqcap \psi) = \sim\!\phi \sqcap \sim\!\psi$	R5
$a \cdot (\sim\!\phi :\rightarrow x) = \phi :\rightarrow a \cdot x + -\phi :\rightarrow a \cdot \delta$	R6

The axioms of $\mathrm{BPA}_\delta^{\mathrm{cr}}$ are the axioms of $\mathrm{BPA}_\delta^{\mathrm{c}}$ and the additional axioms given in Table 18. The crucial axiom is R6, which shows that a conditional expression of the form $\sim\!\zeta :\rightarrow p$ gives a retrospection at the condition under which the immediately preceding action has been performed.

Recall that we write $p \lhd \zeta \rhd q$ for $\zeta :\rightarrow p + -\zeta :\rightarrow q$. An interesting equation is $a \cdot (x \lhd \sim\!\phi \rhd y) = a \cdot x \lhd \phi \rhd a \cdot y$. This equation is a generalization of axiom R6: axiom R6 is derivable from the other axioms of $\mathrm{BPA}_\delta^{\mathrm{cr}}$ and this equation by substituting δ for y and applying axioms GC3 and A6. It is not immediately clear that this equation is derivable from the axioms of $\mathrm{BPA}_\delta^{\mathrm{cr}}$.

Proposition 24 (Derivability Generalization Axiom R6) *The equation* $a \cdot (x \lhd \sim\!\phi \rhd y) = a \cdot x \lhd \phi \rhd a \cdot y$ (R6′) *is derivable from the axioms of* $\mathrm{BPA}_\delta^{\mathrm{cr}}$.

PROOF. We can make the following derivation:

$$\begin{aligned}
&a \cdot (x \lhd \sim\!\phi \rhd y)\\
&\quad\overset{\mathrm{BA2,GC1}}{=} (\phi \sqcup -\phi) :\rightarrow a \cdot (x \lhd \sim\!\phi \rhd y)\\
&\quad\overset{\mathrm{GC7}}{=} \phi :\rightarrow a \cdot (x \lhd \sim\!\phi \rhd y) + -\phi :\rightarrow a \cdot (x \lhd \sim\!\phi \rhd y)\\
&\quad\overset{\mathrm{A6,GC2,BA6,BA,GC6,GC4}}{=} \phi :\rightarrow (\phi :\rightarrow a \cdot (x \lhd \sim\!\phi \rhd y) + -\phi :\rightarrow a \cdot \delta) +\\
&\qquad -\phi :\rightarrow (-\phi :\rightarrow a \cdot (x \lhd \sim\!\phi \rhd y) + \phi :\rightarrow a \cdot \delta)\,.
\end{aligned}$$

Hence, if we can derive $\phi :\rightarrow (\phi :\rightarrow a \cdot (x \lhd \sim\!\phi \rhd y) + -\phi :\rightarrow a \cdot \delta) = \phi :\rightarrow a \cdot x$ (*) and $-\phi :\rightarrow (-\phi :\rightarrow a \cdot (x \lhd \sim\!\phi \rhd y) + \phi :\rightarrow a \cdot \delta) = -\phi :\rightarrow a \cdot y$ (**), then it follows immediately that we can derive R6′. We can derive (*) as follows:

$$\begin{array}{cl}
\multicolumn{2}{l}{\phi :\rightarrow (\phi :\rightarrow a \cdot (x \lhd {\sim}\phi \rhd y) + -\phi :\rightarrow a \cdot \delta)} \\
\overset{\text{R6}}{=} & \phi :\rightarrow a \cdot ({\sim}\phi :\rightarrow (x \lhd {\sim}\phi \rhd y)) \\
\overset{\text{GC4,GC6,BA,BA6,GC2,A6}}{=} & \phi :\rightarrow a \cdot ({\sim}\phi :\rightarrow x) \\
\overset{\text{R6}}{=} & \phi :\rightarrow (\phi :\rightarrow a \cdot x + -\phi :\rightarrow a \cdot \delta) \\
\overset{\text{GC4,GC6,BA,BA6,GC2,A6}}{=} & \phi :\rightarrow a \cdot x \ .
\end{array}$$

We can derive (**) analogously. □

Example 25 *We return to Example 1, which is concerned with a pedestrian who uses a crossing with traffic lights to cross a road with busy traffic safely. Recall that the description of the behaviour of the pedestrian given in Example 1 is as follows:*

$$\begin{array}{l}
\mathit{PED} = \\
\quad \mathit{arrive} \cdot (\mathit{green} :\rightarrow \mathit{cross} + \mathit{red} :\rightarrow (\mathit{make\_req} \cdot (\mathit{green} :\rightarrow \mathit{cross}))) \ .
\end{array}$$

This description concerns a pedestrian who does not act unthinkingly. Now consider a pedestrian who does act unthinkingly. When this pedestrian arrives at the crossing, he or she first makes a request for green light and then crosses the street unconditionally if the light for pedestrians was green on arrival and crosses the street when the light for pedestrians has changed if it was red on arrival. This behaviour can be described in $\mathrm{BPA}_\delta^{\mathrm{cr}}$ *as follows:*

$$\begin{array}{l}
\mathit{PED}' = \\
\quad \mathit{arrive} \cdot \mathit{make\_req} \cdot ({\sim}\mathit{green} :\rightarrow \mathit{cross} + {\sim}\mathit{red} :\rightarrow (\mathit{green} :\rightarrow \mathit{cross})) \ .
\end{array}$$

Because of the addition of the retrospection operator, we cannot use the Boolean algebras $\mathcal{C}_\kappa$ here. The algebras $\mathcal{C}_\kappa^{\mathrm{r}}$ that we use here can be characterized as the free κ-complete algebras over $\mathsf{C_{at}}$ from the class of algebras with interpretations for the constants and operators of Boolean algebras and the retrospection operator that satisfy the axioms of Boolean algebras (Table 1) and axioms R1–R5 from Table 18. We do not make this fully precise, but give an explicit construction of the algebras $\mathcal{C}_\kappa^{\mathrm{r}}$ instead. Important to bear in mind is that not only the atomic conditions, but also the results of applying the operation associated with the retrospection operator a finite number of times to atomic conditions, should not satisfy any equations except those derivable from the axioms.

Let $\mathsf{C}^{\mathrm{r}}_{\mathsf{at}} = \bigcup\{\mathsf{C}_{\mathsf{at}} \times \{i\} \mid i \in \omega\}$ and define $\mathrm{prev} : \mathsf{C}^{\mathrm{r}}_{\mathsf{at}} \to \mathsf{C}^{\mathrm{r}}_{\mathsf{at}}$ by $\mathrm{prev}((\eta, i)) = (\eta, i+1)$. For any infinite cardinal κ, let $\mathcal{C}'_\kappa$ be the free κ-complete Boolean algebra over $\mathsf{C}^{\mathrm{r}}_{\mathsf{at}}$. Then the function prev extends to a unique κ-complete endomorphism prev^* of $\mathcal{C}'_\kappa$. This endomorphism is a unary operation on $\mathcal{C}'_\kappa$ that preserves $\bigsqcup C'$ for every $C' \subseteq \mathcal{C}'_\kappa$ of cardinality less then κ. The algebra $\mathcal{C}^{\mathrm{r}}_\kappa$ is the expansion of $\mathcal{C}'_\kappa$ obtained by associating the operation prev^* with the operator $\sim$. We write $\mathcal{C}^{\mathrm{r}}$ for $\mathcal{C}^{\mathrm{r}}_{\aleph_0}$.

The structural operational semantics of $\mathrm{BPA}^{\mathrm{cr}}_\delta$ differs only from the structural operational semantics of $\mathrm{BPA}^{\mathrm{c}}_\delta$ in the conditions involved. They are now taken from $\mathcal{C}^{\mathrm{r}}$ instead of $\mathcal{C}$.

14 Full Retrospective Splitting Bisimulation Models of $\mathrm{BPA}^{\mathrm{cr}}_\delta$

The construction of the full splitting bisimulation models of $\mathrm{BPA}^{\mathrm{cr}}_\delta$ differs from the construction of the full splitting bisimulation models of $\mathrm{BPA}^{\mathrm{c}}_\delta$ in the conditions involved and in the notion of splitting bisimulation used. The conditions are now taken from $\mathcal{C}^{\mathrm{r}}_\kappa$ instead of $\mathcal{C}_\kappa$. Henceforth, we write $\mathcal{C}^{\mathrm{r}\,-}_\kappa$ for $\mathcal{C}^{\mathrm{r}}_\kappa \setminus \{\bot\}$.

Let κ be an infinite cardinal. Then a *κ-conditional transition system with retrospection* T consists of the following:

- a set S of *states*;
- a set $\xrightarrow{\ell} \subseteq S \times S$, for each $\ell \in \mathcal{C}^{\mathrm{r}\,-}_\kappa \times \mathsf{A}$;
- a set $\xrightarrow{\ell}\surd \subseteq S$, for each $\ell \in \mathcal{C}^{\mathrm{r}\,-}_\kappa \times \mathsf{A}$;
- an *initial state* $s^0 \in S$.

For conditional transition systems with retrospection, reachability, connectedness and connected part are defined exactly as for conditional transition systems.

Let κ be an infinite cardinal. Then $\mathbb{CTS}^{\mathrm{r}}_\kappa$ is the set of all connected κ-conditional transition systems with retrospection $T = (S, \to, \to\surd, s^0)$ such that $S \subset \mathcal{S}_\kappa$ and the branching degree of T is less than κ.

Isomorphism between conditional transition systems with retrospection is defined exactly as for conditional transition systems. Splitting bisimilarity has to be adapted to the setting with retrospection.

Let $T_1 = (S_1, \to_1, \to\surd_1, s^0_1) \in \mathbb{CTS}^{\mathrm{r}}_\kappa$ and $T_2 = (S_2, \to_2, \to\surd_2, s^0_2) \in \mathbb{CTS}^{\mathrm{r}}_\kappa$ (for an infinite cardinal κ). Then a *retrospective splitting bisimulation* B between T_1 and T_2 is a ternary relation $B \subseteq S_1 \times \mathcal{C}^{\mathrm{r}}_\kappa \times S_2$ such that $B(s^0_1, \top, s^0_2)$ and for all s_1, β, s_2 such that $B(s_1, \beta, s_2)$:

- if $s_1 \xrightarrow{[\alpha]\,a}_1 s'_1$, then there is a set $CS'_2 \subseteq \mathcal{C}^{\mathrm{r}\,-}_\kappa \times S_2$ of cardinality less than κ such that $\alpha \sqcap \beta \sqsubseteq \bigsqcup \mathrm{dom}(CS'_2)$ and for all $(\alpha', s'_2) \in CS'_2$, $s_2 \xrightarrow{[\alpha']\,a}_2 s'_2$ and $B(s'_1, {\sim}\alpha', s'_2)$;
- if $s_2 \xrightarrow{[\alpha]\,a}_2 s'_2$, then there is a set $CS'_1 \subseteq \mathcal{C}^{\mathrm{r}\,-}_\kappa \times S_1$ of cardinality less than κ such that $\alpha \sqcap \beta \sqsubseteq \bigsqcup \mathrm{dom}(CS'_1)$ and for all $(\alpha', s'_1) \in CS'_1$, $s_1 \xrightarrow{[\alpha']\,a}_1 s'_1$ and $B(s'_1, {\sim}\alpha', s'_2)$;
- if $s_1 \xrightarrow{[\alpha]\,a} \surd_1$, then there is a set $C' \subseteq \mathcal{C}^{\mathrm{r}\,-}_\kappa$ of cardinality less than κ such that $\alpha \sqcap \beta \sqsubseteq \bigsqcup C'$ and for all $\alpha' \in C'$, $s_2 \xrightarrow{[\alpha']\,a} \surd_2$;
- if $s_2 \xrightarrow{[\alpha]\,a} \surd_2$, then there is a set $C' \subseteq \mathcal{C}^{\mathrm{r}\,-}_\kappa$ of cardinality less than κ such that $\alpha \sqcap \beta \sqsubseteq \bigsqcup C'$ and for all $\alpha' \in C'$, $s_1 \xrightarrow{[\alpha']\,a} \surd_1$.

Two conditional transition systems with retrospection $T_1, T_2 \in \mathbb{CTS}^{\mathrm{r}}_\kappa$ are *retrospective splitting bisimilar*, written $T_1 \mathrel{\underline{\leftrightarrow}}^{\mathrm{r}} T_2$, if there exists a retrospective splitting bisimulation B between T_1 and T_2. Let B be a retrospective splitting bisimulation between T_1 and T_2. Then we say that B is a retrospective splitting bisimulation *witnessing* $T_1 \mathrel{\underline{\leftrightarrow}}^{\mathrm{r}} T_2$.

It is straightforward to see that $\mathrel{\underline{\leftrightarrow}}^{\mathrm{r}}$ is an equivalence on $\mathbb{CTS}^{\mathrm{r}}_\kappa$. Let $T \in \mathbb{CTS}^{\mathrm{r}}_\kappa$. Then we write $[\,T\,]_{\underline{\leftrightarrow}^{\mathrm{r}}}$ for $\{T' \in \mathbb{CTS}^{\mathrm{r}}_\kappa \mid T \mathrel{\underline{\leftrightarrow}}^{\mathrm{r}} T'\}$, i.e. the $\mathrel{\underline{\leftrightarrow}}^{\mathrm{r}}$-equivalence class of T. We write $\mathbb{CTS}^{\mathrm{r}}_\kappa / {\underline{\leftrightarrow}}^{\mathrm{r}}$ for the set of equivalence classes $\{[\,T\,]_{\underline{\leftrightarrow}^{\mathrm{r}}} \mid T \in \mathbb{CTS}^{\mathrm{r}}_\kappa\}$.

The elements of $\mathbb{CTS}^{\mathrm{r}}_\kappa$ and operations on $\mathbb{CTS}^{\mathrm{r}}_\kappa$ to be associated with the constants δ and a (for each $a \in \mathsf{A}$) and the operators $+$, $\cdot$ and $:\rightarrow$ are defined exactly as the elements of $\mathbb{CTS}_\kappa$ and operations on $\mathbb{CTS}_\kappa$ associated with them before.

Below, we show that retrospective splitting bisimilarity is a congruence with respect to alternative composition, sequential composition and guarded command. That leads us to the use of the notion of a layered retrospective splitting bisimulation.

Let $T = (S, \rightarrow, \rightarrow\surd, s^0) \in \mathbb{CTS}^{\mathrm{r}}_\kappa$ (for an infinite cardinal κ). Then the *reachability in n steps* relations of T, one for each $n \in \mathbb{N}$, are the smallest relations $\overset{n}{\twoheadrightarrow} \subseteq S \times S$ such that:

- $s \overset{0}{\twoheadrightarrow} s$;
- if $s \xrightarrow{\ell} s'$ and $s' \overset{n}{\twoheadrightarrow} s''$, then $s \overset{n+1}{\twoheadrightarrow} s''$.

Let $T_1 = (S_1, \rightarrow_1, \rightarrow\surd_1, s^0_1) \in \mathbb{CTS}^{\mathrm{r}}_\kappa$ and $T_2 = (S_2, \rightarrow_2, \rightarrow\surd_2, s^0_2) \in \mathbb{CTS}^{\mathrm{r}}_\kappa$; and let B be a retrospective splitting bisimulation between T_1 and T_2. Then B is called a *layered retrospective splitting bisimulation* if for all $s_1 \in S_1$, $s_2 \in S_2$ and $\alpha \in \mathcal{C}^{\mathrm{r}}_\kappa$, $B(s_1, \alpha, s_2)$ implies that $s^0_1 \overset{n}{\twoheadrightarrow}_1 s_1$ iff $s^0_2 \overset{n}{\twoheadrightarrow}_2 s_2$ for all $n \in \mathbb{N}$.

Lemma 26 (Existence layered retrospective splitting bisimulation)

Let $T_1 = (S_1, \rightarrow_1, \rightarrow\surd_1, s_1^0) \in \mathbb{CTS}_\kappa^{\mathrm{r}}$ and $T_2 = (S_2, \rightarrow_2, \rightarrow\surd_2, s_2^0) \in \mathbb{CTS}_\kappa^{\mathrm{r}}$ (for an infinite cardinal κ). Then $T_1 \mathrel{\underline{\Leftrightarrow}}^{\mathrm{r}} T_2$ implies that there exists a layered retrospective splitting bisimulation witnessing $T_1 \mathrel{\underline{\Leftrightarrow}}^{\mathrm{r}} T_2$.

PROOF. Let B be a retrospective splitting bisimulation witnessing $T_1 \mathrel{\underline{\Leftrightarrow}}^{\mathrm{r}} T_2$. Then we construct a relation B' as follows: $B' = \{(s_1, \alpha, s_2) \mid B(s_1, \alpha, s_2) \land \forall n \in \mathbb{N} \bullet s_1^0 \overset{n}{\twoheadrightarrow}_1 s_1 \Leftrightarrow s_2^0 \overset{n}{\twoheadrightarrow}_2 s_2\}$. It is easy to see that B' is a retrospective splitting bisimulation witnessing $T_1 \mathrel{\underline{\Leftrightarrow}}^{\mathrm{r}} T_2$ as well. □

Proposition 27 (Congruence) *Let κ be an infinite cardinal. Then for all $T_1, T_2, T_1', T_2' \in \mathbb{CTS}_\kappa^{\mathrm{r}}$ and $\alpha \in \mathcal{C}$, $T_1 \mathrel{\underline{\Leftrightarrow}}^{\mathrm{r}} T_1'$ and $T_2 \mathrel{\underline{\Leftrightarrow}}^{\mathrm{r}} T_2'$ imply $T_1 \mathbin{\hat{+}}^{\mathrm{r}} T_2 \mathrel{\underline{\Leftrightarrow}}^{\mathrm{r}} T_1' \mathbin{\hat{+}}^{\mathrm{r}} T_2'$, $T_1 \mathbin{\hat{\cdot}}^{\mathrm{r}} T_2 \mathrel{\underline{\Leftrightarrow}}^{\mathrm{r}} T_1' \mathbin{\hat{\cdot}}^{\mathrm{r}} T_2'$, $\alpha \mathbin{\widehat{:\rightarrow}}^{\mathrm{r}} T_1 \mathrel{\underline{\Leftrightarrow}}^{\mathrm{r}} \alpha \mathbin{\widehat{:\rightarrow}}^{\mathrm{r}} T_1'$.*

PROOF. Let $T_i = (S_i, \rightarrow_i, \rightarrow\surd_i, s_i^0)$ and $T_i' = (S_i', \rightarrow_i', \rightarrow\surd_i', s_i^{0\prime})$ for $i = 1, 2$. Let R_1 and R_2 be layered retrospective splitting bisimulations witnessing $T_1 \mathrel{\underline{\Leftrightarrow}}^{\mathrm{r}} T_1'$ and $T_2 \mathrel{\underline{\Leftrightarrow}}^{\mathrm{r}} T_2'$, respectively. Then we construct relations $R_{\hat{+}^{\mathrm{r}}}$, $R_{\hat{\cdot}^{\mathrm{r}}}$ and $R_{\widehat{:\rightarrow}^{\mathrm{r}}}$ as follows:

- $R_{\hat{+}^{\mathrm{r}}} = (\{(s^0, \top, s^{0\prime})\} \cup \mu_1(R_1) \cup \mu_2(R_2)) \cap (S \times \mathcal{C}_\kappa^{\mathrm{r}} \times S')$, where S and S' are the sets of states of $T_1 \mathbin{\hat{+}}^{\mathrm{r}} T_2$ and $T_1' \mathbin{\hat{+}}^{\mathrm{r}} T_2'$, respectively, and s^0 and $s^{0\prime}$ are the initial states of $T_1 \mathbin{\hat{+}}^{\mathrm{r}} T_2$ and $T_1' \mathbin{\hat{+}}^{\mathrm{r}} T_2'$, respectively;
- $R_{\hat{\cdot}^{\mathrm{r}}} = (\mu_1(R_1) \cup \mu_2(R_2'')) \cap (S \times \mathcal{C}_\kappa^{\mathrm{r}} \times S')$, where S and S' are the sets of states of $T_1 \mathbin{\hat{\cdot}}^{\mathrm{r}} T_2$ and $T_1' \mathbin{\hat{\cdot}}^{\mathrm{r}} T_2'$, respectively, and $R_2'' = \{(s, {\sim}^{n+1}(\alpha) \sqcap \alpha', s') \mid R_2(s, \alpha', s') \land \exists s_1 \in S_1, a \in \mathsf{A} \bullet s_1 \xrightarrow{[\alpha]\, a} \surd_1 \land s_2^0 \overset{n}{\twoheadrightarrow}_2 s\}$;
- $R_{\widehat{:\rightarrow}^{\mathrm{r}}} = R_1'' \cap (S \times \mathcal{C}_\kappa^{\mathrm{r}} \times S')$, where S and S' are the sets of states of $\alpha \mathbin{\widehat{:\rightarrow}}^{\mathrm{r}} T_1$ and $\alpha \mathbin{\widehat{:\rightarrow}}^{\mathrm{r}} T_1'$, respectively, and $R_1'' = \{(s_1^0, \top, s_1^{0\prime})\} \cup \{(s, {\sim}^{n+1}(\alpha) \sqcap \alpha', s') \mid R_1(s, \alpha', s') \land s_1^0 \overset{n+1}{\twoheadrightarrow}_1 s\}$.

Here, we write $\mu_i(R_i)$ for $\{(\mu_i(s), \alpha, \mu_i(s')) \mid R_i(s, \alpha, s')\}$, where μ_i is used to denote both the injection of S_i into $S_1 \uplus S_2$ and the injection of S_i' into $S_1' \uplus S_2'$. The notation ${\sim}^n(\alpha)$ is defined as follows: ${\sim}^0(\alpha) = \alpha$ and ${\sim}^{n+1}(\alpha) = {\sim}({\sim}^n(\alpha))$. Given the definitions of alternative composition, sequential composition and guarded command, it is straightforward to see that $R_{\hat{+}^{\mathrm{r}}}$, $R_{\hat{\cdot}^{\mathrm{r}}}$ and $R_{\widehat{:\rightarrow}^{\mathrm{r}}}$ are retrospective splitting bisimulations witnessing $T_1 \mathbin{\hat{+}}^{\mathrm{r}} T_2 \mathrel{\underline{\Leftrightarrow}}^{\mathrm{r}} T_1' \mathbin{\hat{+}}^{\mathrm{r}} T_2'$, $T_1 \mathbin{\hat{\cdot}}^{\mathrm{r}} T_2 \mathrel{\underline{\Leftrightarrow}}^{\mathrm{r}} T_1' \mathbin{\hat{\cdot}}^{\mathrm{r}} T_2'$ and $\alpha \mathbin{\widehat{:\rightarrow}}^{\mathrm{r}} T_1 \mathrel{\underline{\Leftrightarrow}}^{\mathrm{r}} \alpha \mathbin{\widehat{:\rightarrow}}^{\mathrm{r}} T_1'$, respectively. □

Note that, in the proof of Proposition 27, showing that $R_{\hat{\cdot}^{\mathrm{r}}}$ and $R_{\widehat{:\rightarrow}^{\mathrm{r}}}$ are witnesses needs the assumption that R_1 and R_2 are layered.

The *full retrospective splitting bisimulation models* $\mathfrak{P}_\kappa^{\mathrm{cr}}$ of $\mathrm{BPA}_\delta^{\mathrm{cr}}$, one for each infinite cardinal κ, are the expansions of $\mathcal{C}^{\mathrm{r}}$ whose additional ingredients are defined as follows:

$$\begin{array}{llll}
\mathcal{P} & = \mathbb{CTS}^{\mathrm{r}}_{\kappa}/{\underline{\leftrightarrow}}^{\mathrm{r}}\ , & [\,T_1\,]_{\underline{\leftrightarrow}^{\mathrm{r}}} \mathbin{\widetilde{+}^{\mathrm{r}}} [\,T_2\,]_{\underline{\leftrightarrow}^{\mathrm{r}}} & = [\,T_1 \mathbin{\widehat{+}^{\mathrm{r}}} T_2\,]_{\underline{\leftrightarrow}^{\mathrm{r}}}\ , \\
\widetilde{\delta}^{\mathrm{r}} & = [\,\widehat{\delta}^{\mathrm{r}}\,]_{\underline{\leftrightarrow}^{\mathrm{r}}}\ , & [\,T_1\,]_{\underline{\leftrightarrow}^{\mathrm{r}}} \mathbin{\widetilde{\cdot}^{\mathrm{r}}} [\,T_2\,]_{\underline{\leftrightarrow}^{\mathrm{r}}} & = [\,T_1 \mathbin{\widehat{\cdot}^{\mathrm{r}}} T_2\,]_{\underline{\leftrightarrow}^{\mathrm{r}}}\ , \\
\widetilde{a}^{\mathrm{r}} & = [\,\widehat{a}^{\mathrm{r}}\,]_{\underline{\leftrightarrow}^{\mathrm{r}}}\ , & \alpha \mathbin{\widetilde{:\rightarrow}^{\mathrm{r}}} [\,T_1\,]_{\underline{\leftrightarrow}^{\mathrm{r}}} & = [\,\alpha \mathbin{\widehat{:\rightarrow}^{\mathrm{r}}} T_1\,]_{\underline{\leftrightarrow}^{\mathrm{r}}}\ .
\end{array}$$

Alternative composition, sequential composition and guarded command on $\mathbb{CTS}^{\mathrm{r}}_{\kappa}/{\underline{\leftrightarrow}}^{\mathrm{r}}$ are well-defined because ${\underline{\leftrightarrow}}^{\mathrm{r}}$ is a congruence with respect to the corresponding operations on $\mathbb{CTS}^{\mathrm{r}}_{\kappa}$.

The structures $\mathfrak{P}^{\mathrm{cr}}_{\kappa}$ are models of $\mathrm{BPA}^{\mathrm{cr}}_{\delta}$.

Theorem 28 (Soundness of $\mathbf{BPA}^{\mathrm{cr}}_{\delta}$) *For each infinite cardinal κ, we have $\mathfrak{P}^{\mathrm{cr}}_{\kappa} \models \mathrm{BPA}^{\mathrm{cr}}_{\delta}$.*

PROOF. Because $\mathfrak{P}^{\mathrm{cr}}_{\kappa}$ is an expansion of $\mathcal{C}^{\mathrm{r}}$, and $\mathcal{C}^{\mathrm{r}}$ is an expansion of a Boolean algebra (see the end of Section 13), it is not necessary to show that the axioms of BA are sound. The soundness of all remaining axioms follows straightforwardly from the definitions of the ingredients of $\mathfrak{P}^{\mathrm{cr}}_{\kappa}$. $\square$

15 ACP with Retrospective Conditions

We proceed with adding parallel composition and encapsulation operators to $\mathrm{BPA}^{\mathrm{cr}}_{\delta}$, resulting in $\mathrm{ACP}^{\mathrm{cr}}$.

For $\mathrm{ACP}^{\mathrm{cr}}$, we need the following auxiliary operators:

- the unary *retrospection shift* operator $\Pi^{+} : \mathbf{P} \to \mathbf{P}$;
- for each $n \in \mathbb{N}$, the unary *restricted retrospection shift* operator $\Pi^{+}_{>n} {:} \mathbf{P} \to \mathbf{P}$;
- for each $n \in \mathbb{N}$, the unary *restricted retrospection shift* operator $\Pi^{+}_{>n} : \mathbf{C} \to \mathbf{C}$.

In the parallel composition of two processes, when an action of one of the processes is performed, the retrospections of the other process that are not internal should go one step further. This is accomplished by the retrospection shift operator. The restricted retrospection shift operators, on processes and conditions, are needed for the axiomatization of the retrospection shift operator. The retrospection shift operator Π^{+} is similar to the history pointer shift operator *hps* from [8].

Table 19
Axioms adapted to retrospection ($a \in \mathsf{A}_\delta$)

$a \mathbin{\|\!\lfloor} x = a \cdot \Pi^+(x)$	CM2R
$a \cdot x \mathbin{\|\!\lfloor} y = a \cdot (x \parallel \Pi^+(y))$	CM3R

Table 20
Additional axioms for retrospection ($a \in \mathsf{A}_\delta$, $\eta \in \mathsf{C}_{\mathsf{at}}$)

$\Pi^+(x) = \Pi^+_{>0}(x)$	RS0
$\Pi^+_{>n}(a) = a$	RS1
$\Pi^+_{>n}(a \cdot x) = a \cdot \Pi^+_{>n+1}(x)$	RS2
$\Pi^+_{>n}(x + y) = \Pi^+_{>n}(x) + \Pi^+_{>n}(y)$	RS3
$\Pi^+_{>n}(\phi :\rightarrow x) = \Pi^+_{>n}(\phi) :\rightarrow \Pi^+_{>n}(x)$	RS4
$\Pi^+_{>n}(\bot) = \bot$	RS5
$\Pi^+_{>n}(\top) = \top$	RS6
$\Pi^+_{>n}(\eta) = \eta$	RS7
$\Pi^+_{>n}(-\phi) = -\Pi^+_{>n}(\phi)$	RS8
$\Pi^+_{>n}(\phi \sqcup \psi) = \Pi^+_{>n}(\phi) \sqcup \Pi^+_{>n}(\psi)$	RS9
$\Pi^+_{>n}(\phi \sqcap \psi) = \Pi^+_{>n}(\phi) \sqcap \Pi^+_{>n}(\psi)$	RS10
$\Pi^+_{>0}({\sim}\phi) = {\sim}({\sim}\phi)$	RS11
$\Pi^+_{>n+1}({\sim}\phi) = {\sim}\Pi^+_{>n}(\phi)$	RS12

The axioms of ACP$^{\mathrm{cr}}$ are the axioms of ACP$^{\mathrm{c}}$ with axioms CM2–CM3 replaced by axioms CM2R–CM3R from Table 19, and the additional axioms for retrospection given in Table 20. Axioms CM2R and CM3R show that retrospections are adapted if two processes proceed in parallel. Axioms RS0–RS12 state that this happens as explained above. By means of axioms RS5–RS12, the retrospection shift operators on conditions can be eliminated from all terms of sort **C**.

The structural operational semantics of ACP$^{\mathrm{cr}}$ is described by the transition rules for ACP$^{\mathrm{c}}$ with the first four transition rules for parallel composition and the two transition rules for left merge replaced by the transition rules given in Table 21, and the additional transition rules for retrospection given in Table 22.

Table 21
Transition rules adapted to retrospection

$$\frac{x \xrightarrow{[\phi]\,a} \surd}{x \parallel y \xrightarrow{[\phi]\,a} \Pi^{+}(y)} \qquad \frac{y \xrightarrow{[\phi]\,a} \surd}{x \parallel y \xrightarrow{[\phi]\,a} \Pi^{+}(x)}$$

$$\frac{x \xrightarrow{[\phi]\,a} x'}{x \parallel y \xrightarrow{[\phi]\,a} x' \parallel \Pi^{+}(y)} \qquad \frac{y \xrightarrow{[\phi]\,a} y'}{x \parallel y \xrightarrow{[\phi]\,a} \Pi^{+}(x) \parallel y'}$$

$$\frac{x \xrightarrow{[\phi]\,a} \surd}{x \mathbin{⌊\!\!⌊} y \xrightarrow{[\phi]\,a} \Pi^{+}(y)} \qquad \frac{x \xrightarrow{[\phi]\,a} x'}{x \mathbin{⌊\!\!⌊} y \xrightarrow{[\phi]\,a} x' \parallel \Pi^{+}(y)}$$

Table 22
Additional transition rules for retrospection

$$\frac{x \xrightarrow{[\phi]\,a} \surd}{\Pi^{+}(x) \xrightarrow{[\Pi^{+}_{>0}(\phi)]\,a} \surd} \qquad \frac{x \xrightarrow{[\phi]\,a} x'}{\Pi^{+}(x) \xrightarrow{[\Pi^{+}_{>0}(\phi)]\,a} \Pi^{+}_{>1}(x')}$$

$$\frac{x \xrightarrow{[\phi]\,a} \surd}{\Pi^{+}_{>n}(x) \xrightarrow{[\Pi^{+}_{>n}(\phi)]\,a} \surd} \qquad \frac{x \xrightarrow{[\phi]\,a} x'}{\Pi^{+}_{>n}(x) \xrightarrow{[\Pi^{+}_{>n}(\phi)]\,a} \Pi^{+}_{>n+1}(x')}$$

16 Full Retrospective Splitting Bisimulation Models of ACP$^{\mathrm{cr}}$

In this section, we expand the full retrospective splitting bisimulation models of BPA$_{\delta}^{\mathrm{cr}}$ to ACP$^{\mathrm{cr}}$. The operations on $\mathbb{CTS}^{\mathrm{r}}_{\kappa}$ that we associate with most of the additional operators of ACP$^{\mathrm{cr}}$ call for unfolding of transition systems from $\mathbb{CTS}^{\mathrm{r}}_{\kappa}$.

For the sake of unfolding, it is assumed that, for each infinite cardinal κ, $\mathcal{S}_{\kappa}$ has the following closure property: [11]

$$\text{for all } S \subseteq \mathcal{S}_{\kappa},\ \{\pi \frown \langle s \rangle \mid \pi \in (S \times (\mathcal{C}^{\mathrm{r}}_{\kappa} \times \mathsf{A}))^{*} \land s \in S\} \subseteq \mathcal{S}_{\kappa}\ .$$

We write $\mathrm{P}'(S)$ for the set $\{\pi \frown \langle s \rangle \mid \pi \in (S \times (\mathcal{C}^{\mathrm{r}}_{\kappa} \times \mathsf{A}))^{*} \land s \in S\}$. The function $\# : \mathrm{P}'(S) \to \mathbb{N}$ is defined by

$$\begin{aligned} \#(\langle s \rangle) &= 0\ , \\ \#(\pi \frown \langle s, \ell, s' \rangle) &= \#(\pi \frown \langle s \rangle) + 1\ . \end{aligned}$$

The elements of $\mathrm{P}'(S)$, for an $S \subseteq \mathcal{S}_{\kappa}$, can be looked upon as potential paths of

[11] We write $\langle\,\rangle$ for the empty sequence, $\langle e \rangle$ for the sequence having e as sole element and $\sigma \frown \sigma'$ for the concatenation of sequences σ and σ'; and we use $\langle e_1, \ldots, e_n \rangle$ as a shorthand for $\langle e_1 \rangle \frown \ldots \frown \langle e_n \rangle$.

a κ-conditional transition system with S as set of states. A (non-terminating) path of a transition system $(S, \rightarrow, \rightarrow\surd, s^0) \in \mathbb{CTS}^{\mathrm{r}}_{\kappa}$ is a finite alternating sequence $\langle s_0, \ell_1, s_1, \ldots, \ell_n, s_n \rangle$ of states from S and labels from $\mathcal{C}^{\mathrm{r}}_{\kappa} \times \mathsf{A}$ such that $s_0 = s^0$ and $s_i \xrightarrow{\ell_{i+1}} s_{i+1}$ for all $i < n$. The state s_n is called the state in which the path ends.

Let $T = (S, \rightarrow, \rightarrow\surd, s^0) \in \mathbb{CTS}^{\mathrm{r}}_{\kappa}$. Then the set of *paths* of T, written $\mathrm{P}(T)$, is the smallest subset of $\mathrm{P}'(S)$ such that:

- $\langle s^0 \rangle \in \mathrm{P}(T)$,
- if $\pi \frown \langle s \rangle \in \mathrm{P}(T)$ and $s \xrightarrow{\ell} s'$, then $\pi \frown \langle s, \ell, s' \rangle \in \mathrm{P}(T)$.

In order to unfold a transition system, we need for each state s of the original transition system, for each different path that ends in state s, a different state in the unfolded transition system. The obvious choice is to take the paths concerned as states.

Let $T = (S, \rightarrow, \rightarrow\surd, s^0) \in \mathbb{CTS}^{\mathrm{r}}_{\kappa}$. Then the *unfolding* of T, written $\Upsilon(T)$, is defined as follows:

$$\Upsilon(T) = (S', \rightarrow', \rightarrow\surd', s^{0\prime})\ ,$$

where

$$S' = \mathrm{P}(T)\ ,$$

and for every $\ell \in \mathcal{C}^{\mathrm{r}\,-}_{\kappa} \times \mathsf{A}$:

$$\begin{aligned}
\xrightarrow{\ell}{}' &= \left\{ (\pi \frown \langle s \rangle, \pi \frown \langle s, \ell, s' \rangle) \;\middle|\; \pi \frown \langle s \rangle \in \mathrm{P}(T) \land s \xrightarrow{\ell} s' \right\}\ ,\\
\xrightarrow{\ell}\surd' &= \left\{ \pi \frown \langle s \rangle \;\middle|\; \pi \frown \langle s \rangle \in \mathrm{P}(T) \land s \xrightarrow{\ell}\surd \right\}\ ,\\
s^{0\prime} &= \langle s^0 \rangle\ .
\end{aligned}$$

The functions upd_1 and upd_2 defined next will be used in the definition of parallel composition on $\mathbb{CTS}^{\mathrm{r}}_{\kappa}$ to adapt the retrospection in steps originating from the first operand and the second operand, respectively.

Let $S_1, S_2 \subseteq \mathcal{S}_{\kappa}$. Then the functions $\mathrm{upd}_i : \mathcal{C}^{\mathrm{r}\,-}_{\kappa} \times \mathrm{P}'(S_1 \times S_2) \to \mathcal{C}^{\mathrm{r}\,-}_{\kappa}$, for $i = 1, 2$, are defined by

$$
\begin{aligned}
&\mathrm{upd}_i(\alpha, \langle (s_1, s_2) \rangle) && = \alpha\ , \\
&\mathrm{upd}_i(\alpha, \langle (s_1, s_2), \ell, (s'_1, s'_2) \rangle \frown \pi') && = \mathrm{upd}_i(\alpha, \langle (s'_1, s'_2) \rangle \frown \pi') \text{ if } s_i \neq s'_i\ , \\
&\mathrm{upd}_i(\alpha, \langle (s_1, s_2), \ell, (s'_1, s'_2) \rangle \frown \pi') && = \\
&\qquad\qquad \mathrm{upd}_i\Bigl(\Pi^{+}_{>\#_i(\langle (s'_1, s'_2) \rangle \frown \pi')}(\alpha), \langle (s'_1, s'_2) \rangle \frown \pi'\Bigr) \text{ if } s_i = s'_i\ .
\end{aligned}
$$

where

$$
\begin{aligned}
&\#_i(\langle (s_1, s_2) \rangle) && = 0\ , \\
&\#_i(\langle (s_1, s_2), \ell, (s'_1, s'_2) \rangle \frown \pi') && = \#_i(\langle (s'_1, s'_2) \rangle \frown \pi') + 1 \text{ if } s_i \neq s'_i\ , \\
&\#_i(\langle (s_1, s_2), \ell, (s'_1, s'_2) \rangle \frown \pi') && = \#_i(\langle (s'_1, s'_2) \rangle \frown \pi') \qquad \text{if } s_i = s'_i\ .
\end{aligned}
$$

Henceforth, we write $\mathrm{upd}(\alpha_1, \alpha_2, \pi)$ for $\mathrm{upd}_1(\alpha_1, \pi) \sqcap \mathrm{upd}_2(\alpha_2, \pi)$.

We proceed with expanding the full retrospective splitting bisimulation models of $\mathrm{BPA}^{\mathrm{cr}}_{\delta}$ to $\mathrm{ACP}^{\mathrm{cr}}$.

We associate with the additional operator $\parallel$ an operation $\widehat{\parallel}^{\mathrm{r}}$ on $\mathbb{CTS}^{\mathrm{r}}_{\kappa}$ as follows.

- Let $T_1, T_2 \in \mathbb{CTS}^{\mathrm{r}}_{\kappa}$. Suppose that $\Upsilon(T_i) = (S_i, \rightarrow_i, \rightarrow\!\surd_i, s^0_i)$ for $i = 1, 2$, and $\Upsilon(\Upsilon(T_1) \mathbin{\widehat{\parallel}} \Upsilon(T_2)) = (S, \rightarrow, \rightarrow\!\surd, s^0)$. Then

$$
T_1 \mathbin{\widehat{\parallel}^{\mathrm{r}}} T_2 = (S, \rightarrow', \rightarrow\!\surd', s^0)\ ,
$$

where for every $(\alpha, a) \in \mathcal{C}_\kappa^{\mathrm{r}-} \times \mathsf{A}$:

$$
\begin{aligned}
\xrightarrow{(\alpha,a)}{}' &= \Big\{(\pi \frown \langle (s_1, s_2)\rangle, \pi' \frown \langle (s_1', s_2')\rangle) \;\Big|\; s_1 \neq s_1' \land s_2 = s_2' \land \\
&\qquad \bigvee_{\alpha' \in \mathcal{C}_\kappa^{\mathrm{r}-}} \Big(\pi \frown \langle (s_1, s_2)\rangle \xrightarrow{[\alpha']\, a} \pi' \frown \langle (s_1', s_2')\rangle \land \\
&\qquad\qquad \mathrm{upd}_1(\alpha', \pi \frown \langle (s_1, s_2)\rangle) = \alpha\Big)\Big\} \\
&\cup \Big\{(\pi \frown \langle (s_1, s_2)\rangle, \pi' \frown \langle (s_1', s_2')\rangle) \;\Big|\; s_1 = s_1' \land s_2 \neq s_2' \land \\
&\qquad \bigvee_{\alpha' \in \mathcal{C}_\kappa^{\mathrm{r}-}} \Big(\pi \frown \langle (s_1, s_2)\rangle \xrightarrow{[\alpha']\, a} \pi' \frown \langle (s_1', s_2')\rangle \land \\
&\qquad\qquad \mathrm{upd}_2(\alpha', \pi \frown \langle (s_1, s_2)\rangle) = \alpha\Big)\Big\} \\
&\cup \Big\{(\pi \frown \langle (s_1, s_2)\rangle, \pi' \frown \langle (s_1', s_2')\rangle) \;\Big| \\
&\qquad \bigvee_{\alpha', \beta' \in \mathcal{C}_\kappa^{\mathrm{r}-}, a', b' \in \mathsf{A}} \Big(\pi \frown \langle (s_1, s_2)\rangle \xrightarrow{[\alpha' \sqcap \beta']\, a} \pi' \frown \langle (s_1', s_2')\rangle \land \\
&\qquad\qquad s_1 \xrightarrow{[\alpha']\, a'}_1 s_1' \land s_2 \xrightarrow{[\beta']\, b'}_2 s_2' \land \\
&\qquad\qquad \mathrm{upd}(\alpha', \beta', \pi \frown \langle (s_1, s_2)\rangle) = \alpha \land \\
&\qquad\qquad\qquad a' \mid b' = a\Big)\Big\}\;,
\end{aligned}
$$

$$
\begin{aligned}
\xrightarrow{(\alpha,a)} \surd' &= \Big\{\pi \frown \langle (s_1, s_2)\rangle \;\Big|\; s_2 \notin S_2 \land \\
&\qquad \bigvee_{\alpha' \in \mathcal{C}_\kappa^{\mathrm{r}-}} \Big(\pi \frown \langle (s_1, s_2)\rangle \xrightarrow{[\alpha']\, a} \surd \land \\
&\qquad\qquad \mathrm{upd}_1(\alpha', \pi \frown \langle (s_1, s_2)\rangle) = \alpha\Big)\Big\} \\
&\cup \Big\{\pi \frown \langle (s_1, s_2)\rangle \;\Big|\; s_1 \notin S_1 \land \\
&\qquad \bigvee_{\alpha' \in \mathcal{C}_\kappa^{\mathrm{r}-}} \Big(\pi \frown \langle (s_1, s_2)\rangle \xrightarrow{[\alpha']\, a} \surd \land \\
&\qquad\qquad \mathrm{upd}_2(\alpha', \pi \frown \langle (s_1, s_2)\rangle) = \alpha\Big)\Big\} \\
&\cup \Big\{\pi \frown \langle (s_1, s_2)\rangle \;\Big| \\
&\qquad \bigvee_{\alpha', \beta' \in \mathcal{C}_\kappa^{\mathrm{r}-}, a', b' \in \mathsf{A}} \Big(\pi \frown \langle (s_1, s_2)\rangle \xrightarrow{[\alpha' \sqcap \beta']\, a} \surd \land \\
&\qquad\qquad s_1 \xrightarrow{[\alpha']\, a'} \surd_1 \land s_2 \xrightarrow{[\beta']\, b'} \surd_2 \land \\
&\qquad\qquad \mathrm{upd}(\alpha', \beta', \pi \frown \langle (s_1, s_2)\rangle) = \alpha \land \\
&\qquad\qquad\qquad a' \mid b' = a\Big)\Big\}\;.
\end{aligned}
$$

Remark 29 *The operation $\widehat{\parallel}^{\mathrm{r}}$ on $\mathbb{CTS}_\kappa^{\mathrm{r}}$ is defined above in a step-by-step way. The basic idea behind this definition is twofold:*

- *$T_1 \mathbin{\widehat{\parallel}^{\mathrm{r}}} T_2$ can be obtained by first composing T_1 and T_2 to $T_1 \mathbin{\widehat{\parallel}} T_2$ and then adapting the retrospections in steps of $T_1 \mathbin{\widehat{\parallel}} T_2$;*
- *unfolding of $T_1 \mathbin{\widehat{\parallel}} T_2$ is needed before the actual adaptations can take place because the adaptation of the retrospection in a step may be different for the different paths that end in the state from which the step starts.*

Somewhat surprisingly, in addition, T_1 and T_2 must be unfolded before the actual composition takes place. In a step where an action of T_1 and an action of T_2 are performed synchronously, the condition under which the action of T_1 can be performed and the condition under which the action of T_2 can be performed are needed to adapt the retrospection in that step correctly. If T_1 and T_2 are not unfolded before the actual composition takes place, in general, those conditions cannot be determined uniquely.

The operations on $\mathbb{CTS}^{\mathrm{r}}_{\kappa}$ to be associated with the additional operators $\lfloor\!\lfloor$ and $|$ are defined analogously. The operations on $\mathbb{CTS}^{\mathrm{r}}_{\kappa}$ to be associated with the additional operators ∂_H are defined exactly as the operations on $\mathbb{CTS}_{\kappa}$ associated with them before. We associate with the additional operators $\Pi^{+}_{>n}$ operations $\widehat{\Pi^{+}_{>n}}^{\mathrm{r}}$ on $\mathbb{CTS}^{\mathrm{r}}_{\kappa}$ as follows.

- Let $T \in \mathbb{CTS}^{\mathrm{r}}_{\kappa}$. Suppose that $\Upsilon(T) = (S, \rightarrow, \rightarrow\surd, s^0)$. Then

$$\widehat{\Pi^{+}_{>n}}^{\mathrm{r}}(T) = (S, \rightarrow', \rightarrow\surd', s^0)\;,$$

  where for every $(\alpha, a) \in \mathcal{C}^{\mathrm{r}\,-}_{\kappa} \times \mathsf{A}$:

$$\begin{aligned}\xrightarrow{(\alpha,a)}{}' &= \Big\{(\pi \frown \langle s\rangle, \pi' \frown \langle s'\rangle) \;\Big|\\ &\qquad \bigvee_{\alpha' \in \mathcal{C}^{\mathrm{r}\,-}_{\kappa}} \Big(\pi \frown \langle s\rangle \xrightarrow{[\alpha']\,a} \pi' \frown \langle s'\rangle \land \Pi^{+}_{>\#(\pi)+n}(\alpha') = \alpha\Big)\Big\}\;,\\ \xrightarrow{(\alpha,a)}\surd' &= \Big\{\pi \frown \langle s\rangle \;\Big|\\ &\qquad \bigvee_{\alpha' \in \mathcal{C}^{\mathrm{r}\,-}_{\kappa}} \Big(\pi \frown \langle s\rangle \xrightarrow{[\alpha']\,a} \surd \land \Pi^{+}_{>\#(\pi)+n}(\alpha') = \alpha\Big)\Big\}\;.\end{aligned}$$

The operation on $\mathbb{CTS}^{\mathrm{r}}_{\kappa}$ to be associated with the additional operator Π^{+} is the same as the operation on $\mathbb{CTS}^{\mathrm{r}}_{\kappa}$ associated with $\Pi^{+}_{>0}$.

We can also show that retrospective splitting bisimilarity is a congruence with respect to parallel composition, left merge, communication merge, encapsulation, retrospection shift and restricted retrospection shift.

Proposition 30 (Congruence) *Let κ be an infinite cardinal. Then for all $T_1, T_2, T_1', T_2' \in \mathbb{CTS}^{\mathrm{r}}_{\kappa}$, $T_1 \mathrel{\underline{\leftrightarrow}}^{\mathrm{r}} T_1'$ and $T_2 \mathrel{\underline{\leftrightarrow}}^{\mathrm{r}} T_2'$ imply $T_1 \mathbin{\widehat{\|}^{\mathrm{r}}} T_2 \mathrel{\underline{\leftrightarrow}}^{\mathrm{r}} T_1' \mathbin{\widehat{\|}^{\mathrm{r}}} T_2'$, $T_1 \mathbin{\widehat{\lfloor\!\lfloor}^{\mathrm{r}}} T_2 \mathrel{\underline{\leftrightarrow}}^{\mathrm{r}} T_1' \mathbin{\widehat{\lfloor\!\lfloor}^{\mathrm{r}}} T_2'$, $T_1 \mathbin{\widehat{|}^{\mathrm{r}}} T_2 \mathrel{\underline{\leftrightarrow}}^{\mathrm{r}} T_1' \mathbin{\widehat{|}^{\mathrm{r}}} T_2'$, $\widehat{\partial_H}^{\mathrm{r}}(T_1) \mathrel{\underline{\leftrightarrow}}^{\mathrm{r}} \widehat{\partial_H}^{\mathrm{r}}(T_1')$, $\widehat{\Pi^{+}}^{\mathrm{r}}(T_1) \mathrel{\underline{\leftrightarrow}}^{\mathrm{r}} \widehat{\Pi^{+}}^{\mathrm{r}}(T_1')$ and $\widehat{\Pi^{+}_{>n}}^{\mathrm{r}}(T_1) \mathrel{\underline{\leftrightarrow}}^{\mathrm{r}} \widehat{\Pi^{+}_{>n}}^{\mathrm{r}}(T_1')$.*

PROOF. It is easy to see that, for all $T, T' \in \mathbb{CTS}^{\mathrm{r}}_{\kappa}$, $T \mathrel{\underline{\leftrightarrow}}^{\mathrm{r}} T'$ implies $\Upsilon(T) \mathrel{\underline{\leftrightarrow}}^{\mathrm{r}} \Upsilon(T')$. Hence, $\Upsilon(T_1) \mathrel{\underline{\leftrightarrow}}^{\mathrm{r}} \Upsilon(T_1')$ and $\Upsilon(T_2) \mathrel{\underline{\leftrightarrow}}^{\mathrm{r}} \Upsilon(T_2')$. Let R_1' and R_2' be retrospective splitting bisimulations witnessing $\Upsilon(T_1) \mathrel{\underline{\leftrightarrow}}^{\mathrm{r}} \Upsilon(T_1')$ and $\Upsilon(T_2) \mathrel{\underline{\leftrightarrow}}^{\mathrm{r}} \Upsilon(T_2')$, respectively; and let R_1 be a layered retrospective splitting bisimulation witnessing $T_1 \mathrel{\underline{\leftrightarrow}}^{\mathrm{r}} T'_1$. Then we construct relations $R_{\widehat{\|}^{\mathrm{r}}}$, $R_{\widehat{\lfloor\!\lfloor}^{\mathrm{r}}}$, $R_{\widehat{\mid}^{\mathrm{r}}}$, $R_{\widehat{\partial_H}^{\mathrm{r}}}$, $R_{\widehat{\Pi^+}^{\mathrm{r}}}$ and $R_{\widehat{\Pi^+_{>n}}^{\mathrm{r}}}$ as follows:

- Let S and S' be the sets of states of $T_1 \mathbin{\widehat{\|}^{\mathrm{r}}} T_2$ and $T'_1 \mathbin{\widehat{\|}^{\mathrm{r}}} T'_2$, respectively, and let s^0 and $s^{0\prime}$ be the initial states of $T_1 \mathbin{\widehat{\|}^{\mathrm{r}}} T_2$ and $T'_1 \mathbin{\widehat{\|}^{\mathrm{r}}} T'_2$, respectively. Then $R_{\widehat{\|}^{\mathrm{r}}}$ is the smallest subset of $S \times \mathcal{C}^{\mathrm{r}}_{\kappa} \times S'$ such that:
  - $R_{\widehat{\|}^{\mathrm{r}}}(s^0, \top, s^{0\prime})$;
  - if $R_{\widehat{\|}^{\mathrm{r}}}(\pi \frown \langle(\pi_1,\pi_2)\rangle, \alpha, \pi' \frown \langle(\pi'_1,\pi'_2)\rangle)$,
    $R'_1(\pi''_1 \frown \langle s_1\rangle, \alpha_1, \pi'''_1 \frown \langle s'_1\rangle)$,
    $R'_2(\pi''_2 \frown \langle s_2\rangle, \alpha_2, \pi'''_2 \frown \langle s'_2\rangle)$,
    $\pi \frown \langle(\pi_1,\pi_2), \ell, (\pi''_1 \frown \langle s_1\rangle, \pi''_2 \frown \langle s_2\rangle)\rangle \in S$,
    $\pi' \frown \langle(\pi'_1,\pi'_2), \ell, (\pi'''_1 \frown \langle s'_1\rangle, \pi'''_2 \frown \langle s'_2\rangle)\rangle \in S'$,
    $((s_1 \neq s'_1 \land s_2 = s'_2 \land \mathrm{upd}_1(\alpha_1, \pi \frown \langle(\pi_1,\pi_2)\rangle) = \alpha') \lor$
    $(s_1 = s'_1 \land s_2 \neq s'_2 \land \mathrm{upd}_2(\alpha_2, \pi \frown \langle(\pi_1,\pi_2)\rangle) = \alpha') \lor$
    $(s_1 \neq s'_1 \land s_2 \neq s'_2 \land \mathrm{upd}(\alpha_1, \alpha_2, \pi \frown \langle(\pi_1,\pi_2)\rangle) = \alpha'))$,
    then $R_{\widehat{\|}^{\mathrm{r}}}(\pi \frown \langle(\pi_1,\pi_2), \ell, (\pi''_1 \frown \langle s_1\rangle, \pi''_2 \frown \langle s_2\rangle)\rangle, \alpha',$
    $\pi' \frown \langle(\pi'_1,\pi'_2), \ell, (\pi'''_1 \frown \langle s'_1\rangle, \pi'''_2 \frown \langle s'_2\rangle)\rangle)$.
- $R_{\widehat{\lfloor\!\lfloor}^{\mathrm{r}}}$ is constructed analogous to $R_{\widehat{\|}^{\mathrm{r}}}$.
- $R_{\widehat{\mid}^{\mathrm{r}}}$ is constructed analogous to $R_{\widehat{\|}^{\mathrm{r}}}$.
- $R_{\widehat{\partial_H}^{\mathrm{r}}} = R_1 \cap (S \times S')$, where S and S' are the sets of states of $\widehat{\partial_H}^{\mathrm{r}}(T_1)$ and $\widehat{\partial_H}^{\mathrm{r}}(T'_1)$, respectively.
- $R_{\widehat{\Pi^+}^{\mathrm{r}}} = \{(\pi \frown \langle s\rangle, \Pi^+_{>\#(\pi)}(\alpha), \pi' \frown \langle s'\rangle) \in S \times {\mathcal{C}^{\mathrm{r}}_{\kappa}}^{-} \times S' \mid R_1(s,\alpha,s')\}$, where S and S' are the sets of states of $\widehat{\Pi^+}^{\mathrm{r}}(T_1)$ and $\widehat{\Pi^+}^{\mathrm{r}}(T'_1)$, respectively.
- $R_{\widehat{\Pi^+_{>n}}^{\mathrm{r}}} = \{(\pi \frown \langle s\rangle, \Pi^+_{>\#(\pi)+n}(\alpha), \pi' \frown \langle s'\rangle) \in S \times {\mathcal{C}^{\mathrm{r}}_{\kappa}}^{-} \times S' \mid R_1(s,\alpha,s')\}$, where S and S' are the sets of states of $\widehat{\Pi^+_{>n}}^{\mathrm{r}}(T_1)$ and $\widehat{\Pi^+_{>n}}^{\mathrm{r}}(T'_1)$, respectively.

Given the definitions of parallel composition, left merge, communication merge, encapsulation, retrospection shift and restricted retrospection shift, it is straightforward to see that $R_{\widehat{\|}^{\mathrm{r}}}$, $R_{\widehat{\lfloor\!\lfloor}^{\mathrm{r}}}$, $R_{\widehat{\mid}^{\mathrm{r}}}$, $R_{\widehat{\partial_H}^{\mathrm{r}}}$, $R_{\widehat{\Pi^+}^{\mathrm{r}}}$ and $R_{\widehat{\Pi^+_{>n}}^{\mathrm{r}}}$ are retrospective splitting bisimulations witnessing $T_1 \mathbin{\widehat{\|}^{\mathrm{r}}} T_2 \mathrel{\underline{\leftrightarrow}}^{\mathrm{r}} T'_1 \mathbin{\widehat{\|}^{\mathrm{r}}} T'_2$, $T_1 \mathbin{\widehat{\lfloor\!\lfloor}^{\mathrm{r}}} T_2 \mathrel{\underline{\leftrightarrow}}^{\mathrm{r}} T'_1 \mathbin{\widehat{\lfloor\!\lfloor}^{\mathrm{r}}} T'_2$, $T_1 \mathbin{\widehat{\mid}^{\mathrm{r}}} T_2 \mathrel{\underline{\leftrightarrow}}^{\mathrm{r}} T'_1 \mathbin{\widehat{\mid}^{\mathrm{r}}} T'_2$, $\widehat{\partial_H}^{\mathrm{r}}(T_1) \mathrel{\underline{\leftrightarrow}}^{\mathrm{r}} \widehat{\partial_H}^{\mathrm{r}}(T'_1)$, $\widehat{\Pi^+}^{\mathrm{r}}(T_1) \mathrel{\underline{\leftrightarrow}}^{\mathrm{r}} \widehat{\Pi^+}^{\mathrm{r}}(T'_1)$ and $\widehat{\Pi^+_{>n}}^{\mathrm{r}}(T_1) \mathrel{\underline{\leftrightarrow}}^{\mathrm{r}} \widehat{\Pi^+_{>n}}^{\mathrm{r}}(T'_1)$, respectively. □

Note that, in the proof of Proposition 30, showing that $R_{\widehat{\Pi^+}^{\mathrm{r}}}$ and $R_{\widehat{\Pi^+_{>n}}^{\mathrm{r}}}$ are witnesses needs the assumption that R_1 is layered.

The *full retrospective splitting bisimulation models* $\mathfrak{P}^{\mathrm{cr}\prime}_{\kappa}$ of ACP$^{\mathrm{cr}}$, one for each infinite cardinal κ, are the expansions of the full retrospective splitting bisimulation models $\mathfrak{P}^{\mathrm{cr}}_{\kappa}$ of BPA$^{\mathrm{cr}}_{\delta}$ with an operation $\widetilde{f}^{\mathrm{r}}$ on $\mathbb{CTS}^{\mathrm{r}}_{\kappa}/\underline{\leftrightarrow}^{\mathrm{r}}$ for each additional operator f of ACP$^{\mathrm{cr}}$. Those additional operations are defined as follows:

$$\begin{array}{llll}
[\,T_1\,]_{\underline{\leftrightarrow}^{\mathrm{r}}} \mathbin{\widetilde{\|}^{\mathrm{r}}} [\,T_2\,]_{\underline{\leftrightarrow}^{\mathrm{r}}} = [\,T_1 \mathbin{\widehat{\|}^{\mathrm{r}}} T_2\,]_{\underline{\leftrightarrow}^{\mathrm{r}}}\;, & & \widetilde{\partial_H}^{\mathrm{r}}([\,T_1\,]_{\underline{\leftrightarrow}^{\mathrm{r}}}) & = [\,\widehat{\partial_H}^{\mathrm{r}}(T_1)\,]_{\underline{\leftrightarrow}^{\mathrm{r}}}\;, \\
[\,T_1\,]_{\underline{\leftrightarrow}^{\mathrm{r}}} \mathbin{\widetilde{\lfloor\!\lfloor}^{\mathrm{r}}} [\,T_2\,]_{\underline{\leftrightarrow}^{\mathrm{r}}} = [\,T_1 \mathbin{\widehat{\lfloor\!\lfloor}^{\mathrm{r}}} T_2\,]_{\underline{\leftrightarrow}^{\mathrm{r}}}\;, & & \widetilde{\Pi^{+}}^{\mathrm{r}}([\,T_1\,]_{\underline{\leftrightarrow}^{\mathrm{r}}}) & = [\,\widehat{\Pi^{+}}^{\mathrm{r}}(T_1)\,]_{\underline{\leftrightarrow}^{\mathrm{r}}}\;, \\
[\,T_1\,]_{\underline{\leftrightarrow}^{\mathrm{r}}} \mathbin{\widetilde{|}^{\mathrm{r}}} [\,T_2\,]_{\underline{\leftrightarrow}^{\mathrm{r}}} = [\,T_1 \mathbin{\widehat{|}^{\mathrm{r}}} T_2\,]_{\underline{\leftrightarrow}^{\mathrm{r}}}\;, & & \widetilde{\Pi^{+}_{>n}}^{\mathrm{r}}([\,T_1\,]_{\underline{\leftrightarrow}^{\mathrm{r}}}) & = [\,\widehat{\Pi^{+}_{>n}}^{\mathrm{r}}(T_1)\,]_{\underline{\leftrightarrow}^{\mathrm{r}}}\;.
\end{array}$$

Parallel composition, left merge, communication merge, encapsulation, retrospection shift and restricted retrospection shift on $\mathbb{CTS}^{\mathrm{r}}_{\kappa}/\underline{\leftrightarrow}^{\mathrm{r}}$ are well-defined because $\underline{\leftrightarrow}^{\mathrm{r}}$ is a congruence with respect to the corresponding operations on $\mathbb{CTS}^{\mathrm{r}}_{\kappa}$.

The structures $\mathfrak{P}^{\mathrm{cr}\prime}_{\kappa}$ are models of ACP$^{\mathrm{cr}}$.

Theorem 31 (Soundness of ACP$^{\mathrm{cr}}$) *For each infinite cardinal κ, we have $\mathfrak{P}^{\mathrm{cr}\prime}_{\kappa} \models$ ACP$^{\mathrm{cr}}$.*

PROOF. Because $\mathfrak{P}^{\mathrm{cr}\prime}_{\kappa}$ is an expansion of $\mathfrak{P}^{\mathrm{cr}}_{\kappa}$, it is sufficient to show that the additional axioms for ACP$^{\mathrm{cr}}$ are sound. The soundness of all additional axioms follows straightforwardly from the definitions of the ingredients of $\mathfrak{P}^{\mathrm{cr}\prime}_{\kappa}$. $\square$

Above, the full retrospective splitting bisimulation models $\mathfrak{P}^{\mathrm{cr}}_{\kappa}$ of BPA$^{\mathrm{cr}}_{\delta}$ have been expanded to obtain the full retrospective splitting bisimulation models $\mathfrak{P}^{\mathrm{cr}\prime}_{\kappa}$ of ACP$^{\mathrm{cr}}$. We will loosely write $\mathfrak{P}^{\mathrm{cr}}_{\kappa}$ for $\mathfrak{P}^{\mathrm{cr}\prime}_{\kappa}$.

In the full retrospective splitting bisimulation models of ACP$^{\mathrm{cr}}$, guarded recursive specifications over ACP$^{\mathrm{cr}}$ have unique solutions.

Theorem 32 (Unique solutions in $\mathfrak{P}^{\mathrm{cr}}_{\kappa}$) *For each infinite cardinal κ, guarded recursive specifications over* ACP$^{\mathrm{cr}}$ *have unique solutions in $\mathfrak{P}^{\mathrm{cr}}_{\kappa}$.*

PROOF. The proof follows the same line as the proof of Theorem 10. Here, it is crucial that it is straightforward to define a normal form of elements of $\mathbb{CTS}^{\mathrm{r}}_{\kappa}$ such that: (a) each element of $\mathbb{CTS}^{\mathrm{r}}_{\kappa}$ is retrospective splitting bisimilar to its normal form and (b) two elements of $\mathbb{CTS}^{\mathrm{r}}_{\kappa}$ are retrospective splitting bisimilar iff their normal forms are splitting bisimilar. $\square$

Thus, the full retrospective splitting bisimulation models $\mathfrak{P}^{\mathrm{cr}\prime\prime}_{\kappa}$ of ACP$^{\mathrm{cr}}$ with guarded recursion are simply the expansions of the full retrospective splitting bisimulation models $\mathfrak{P}^{\mathrm{cr}}_{\kappa}$ of ACP$^{\mathrm{cr}}$ obtained by associating with each constant $\langle X|E\rangle$ the unique solution of E for X in the full retrospective splitting bisimulation model concerned.

17 Evaluation of Retrospective Conditions

In this section, we add condition evaluation operators and generalized condition evaluation operators to ACP$^{\mathrm{cr}}$. As in the case of ACP$^{\mathrm{c}}$, these operators require to fix an infinite cardinal λ. By doing so, full retrospective splitting bisimulation models with process domain $\mathbb{CTS}^{\mathrm{r}}_{\kappa}/{\underline{\leftrightarrow}}^{\mathrm{r}}$ for $\kappa > \lambda$ are excluded.

Henceforth, we write $\mathcal{H}^{\mathrm{r}}_{\lambda}$ for the set of all λ-complete endomorphisms of $\mathcal{C}^{\mathrm{r}}_{\lambda}$.

In the case of ACP$^{\mathrm{cr}}$, there are λ-complete condition evaluation operators $\mathsf{CE}_h{:}\mathbf{P} \to \mathbf{P}$ and $\mathsf{CE}_h{:}\mathbf{C} \to \mathbf{C}$, and generalized λ-complete condition evaluation operators $\mathsf{GCE}_h : \mathbf{P} \to \mathbf{P}$ and $\mathsf{GCE}_h : \mathbf{C} \to \mathbf{C}$, for each $h \in \mathcal{H}^{\mathrm{r}}_{\lambda}$. We also need the following auxiliary operators:

- for each $h \in \mathcal{H}^{\mathrm{r}}_{\lambda}$, $n \in \mathbb{N}$, the unary *retrospection update* operator $\Pi^h_n{:}\mathbf{P} \to \mathbf{P}$;
- for each $h \in \mathcal{H}^{\mathrm{r}}_{\lambda}$, $n \in \mathbb{N}$, the unary *retrospection update* operator $\Pi^h_n{:}\mathbf{C} \to \mathbf{C}$.

In the case of ACP$^{\mathrm{cr}}$, it is assumed that a fixed but arbitrary function $\mathsf{eff} : \mathsf{A} \times \mathcal{H}^{\mathrm{r}}_{\lambda} \to \mathcal{H}^{\mathrm{r}}_{\lambda}$ has been given. The function eff is extended to A_{δ} such that $\mathsf{eff}(\delta, h) = h$ for all $h \in \mathcal{H}^{\mathrm{r}}_{\lambda}$.

The condition evaluation operators and generalized condition evaluation operators cannot be added to ACP$^{\mathrm{cr}}$ in the same way as they are added to ACP$^{\mathrm{c}}$. First of all, retrospective conditions may refer back too far to be evaluated. The effect is that, in condition evaluation or generalized condition evaluation of a process according to some endomorphism, the retrospective conditions that refer back further than the beginning of the process have to be left unevaluated. This is accomplished by the retrospection update operators mentioned above. In the case of generalized condition evaluation, there is another complication. Recall that generalized condition evaluation allows the results of condition evaluation to change by performing an action. In the presence of retrospection, different parts of a condition may have to be evaluated differently because of such changes. The effect is that, in generalized condition evaluation of a process according to some endomorphism, after an action of the process is performed, the subsequent retrospective conditions that refer back to the beginning of the process have to be evaluated according to that endomorphism as well. This is also accomplished by the retrospection update

Table 23
New axioms for (generalized) condition evaluation ($a \in \mathsf{A}_\delta$)

$\mathsf{CE}_h(a) = a$	CE1
$\mathsf{CE}_h(a \cdot x) = a \cdot \mathsf{CE}_h(\Pi^h_1(x))$	CE2R
$\mathsf{CE}_h(x + y) = \mathsf{CE}_h(x) + \mathsf{CE}_h(y)$	CE3
$\mathsf{CE}_h(\phi :\to x) = \Pi^h_0(\phi) :\to \mathsf{CE}_h(x)$	CE4R
$\mathsf{GCE}_h(a) = a$	GCE1
$\mathsf{GCE}_h(a \cdot x) = a \cdot \mathsf{GCE}_{\mathsf{eff}(a,h)}(\Pi^h_1(x))$	GCE2R
$\mathsf{GCE}_h(x + y) = \mathsf{GCE}_h(x) + \mathsf{GCE}_h(y)$	GCE3
$\mathsf{GCE}_h(\phi :\to x) = \Pi^h_0(\phi) :\to \mathsf{GCE}_h(x)$	GCE4R

Table 24
Axioms for retrospection update ($a \in \mathsf{A}_\delta$, $\eta \in \mathsf{C}_{\mathsf{at}}$, $\eta' \in \mathsf{C}_{\mathsf{at}} \cup \{\bot, \top\}$)

$\Pi^h_n(a) = a$	RU1	$\Pi^h_0(\eta) = \eta'$	if $h(\eta) = \eta'$	RU7
$\Pi^h_n(a \cdot x) = a \cdot \Pi^h_{n+1}(x)$	RU2	$\Pi^h_{n+1}(\eta) = \eta$		RU8
$\Pi^h_n(x + y) = \Pi^h_n(x) + \Pi^h_n(y)$	RU3	$\Pi^h_n(-\phi) = -\Pi^h_n(\phi)$		RU9
$\Pi^h_n(\phi :\to x) = \Pi^h_n(\phi) :\to \Pi^h_n(x)$	RU4	$\Pi^h_n(\phi \sqcup \psi) = \Pi^h_n(\phi) \sqcup \Pi^h_n(\psi)$		RU10
		$\Pi^h_n(\phi \sqcap \psi) = \Pi^h_n(\phi) \sqcap \Pi^h_n(\psi)$		RU11
$\Pi^h_n(\bot) = \bot$	RU5	$\Pi^h_0({\sim}\phi) = {\sim}\phi$		RU12
$\Pi^h_n(\top) = \top$	RU6	$\Pi^h_{n+1}({\sim}\phi) = {\sim}\Pi^h_n(\phi)$		RU13

operators mentioned above.

For a clear understanding of the retrospection update operators, the following additional remarks are in order. By merely evaluating ahead as described above, in the end only the retrospective conditions that refer back further than the beginning of the process are unevaluated. In other words, by dealing with the second complication, the first complication is dealt with as well, except for the conditions occurring at the beginning of the process. Even if there is no necessity for evaluating ahead because of the second complication, it still deals properly with the first complication.

In the case of ACP$^{\mathrm{cr}}$, the additional axioms for CE_h and GCE_h, where $h \in \mathcal{H}^{\mathrm{r}}_\lambda$, are the axioms given in Tables 23 and 24. These additional axioms differ from the additional axioms in the absence of retrospection (Tables 8 and 10) in that axioms CE2, CE4, GCE2 and GCE4 have been replaced by axioms CE2R, CE4R, GCE2R and GCE4R, and axioms CE6–CE11 by axioms RU1–RU13. Axioms CE2R, CE4R, GCE2R, GCE4R and RU1–RU13 state that condition evaluation and generalized condition evaluation take place as explained above.

Table 25
New transition rules for (generalized) condition evaluation

$$\frac{x \xrightarrow{[\phi]\,a} \surd}{\mathsf{CE}_h(x) \xrightarrow{[\Pi^h_0(\phi)]\,a} \surd}\ \Pi^h_0(\phi) \neq \bot \qquad \frac{x \xrightarrow{[\phi]\,a} x'}{\mathsf{CE}_h(x) \xrightarrow{[\Pi^h_0(\phi)]\,a} \mathsf{CE}_h(\Pi^h_1(x'))}\ \Pi^h_0(\phi) \neq \bot$$

$$\frac{x \xrightarrow{[\phi]\,a} \surd}{\mathsf{GCE}_h(x) \xrightarrow{[\Pi^h_0(\phi)]\,a} \surd}\ \Pi^h_0(\phi) \neq \bot$$

$$\frac{x \xrightarrow{[\phi]\,a} x'}{\mathsf{GCE}_h(x) \xrightarrow{[\Pi^h_0(\phi)]\,a} \mathsf{GCE}_{\mathsf{eff}(a,h)}(\Pi^h_1(x'))}\ \Pi^h_0(\phi) \neq \bot$$

$$\frac{x \xrightarrow{[\phi]\,a} \surd}{\Pi^h_n(x) \xrightarrow{[\Pi^h_n(\phi)]\,a} \surd}\ \Pi^h_n(\phi) \neq \bot \qquad \frac{x \xrightarrow{[\phi]\,a} x'}{\Pi^h_n(x) \xrightarrow{[\Pi^h_n(\phi)]\,a} \Pi^h_{n+1}(x')}\ \Pi^h_n(\phi) \neq \bot$$

Example 33 *We return to Example 25, which is concerned with a pedestrian who uses a crossing with traffic lights to cross a road with busy traffic safely, but acts unthinkingly. Recall that the description of the behaviour of the unthinkingly acting pedestrian given in Example 25 is as follows:*

$$\begin{aligned}&\mathit{PED}' = \\ &\quad \mathit{arrive} \cdot \mathit{make\_req} \cdot ({\sim}\mathit{green} :\rightarrow \mathit{cross} + {\sim}\mathit{red} :\rightarrow (\mathit{green} :\rightarrow \mathit{cross}))\ .\end{aligned}$$

Like in Example 15, let h_g be such that $h_g(\mathit{green}) = \top$ and $h_g(\mathit{red}) = \bot$, let h_r be such that $h_r(\mathit{green}) = \bot$ and $h_r(\mathit{red}) = \top$, and let eff *be such that* $\mathsf{eff}(\mathit{make\_req}, h_r) = h_g$ *and* $\mathsf{eff}(a, h) = h$ *otherwise. Then we can derive the following:*

$$\begin{aligned}&\mathsf{GCE}_{h_g}(\mathit{PED}') = \mathit{arrive} \cdot \mathit{make\_req} \cdot \mathit{cross}\ , \\ &\mathsf{GCE}_{h_r}(\mathit{PED}') = \mathit{arrive} \cdot \mathit{make\_req} \cdot \mathit{cross}\ .\end{aligned}$$

As to be expected, the unthinkingly acting pedestrian will act the same regardless the color of the traffic light for pedestrians on arrival.

The structural operational semantics of $\mathrm{ACP^{cr}}$ extended with condition evaluation and generalized condition evaluation is described by the transition rules for $\mathrm{ACP^{cr}}$ and the transition rules given in Table 25.

The full retrospective splitting bisimulation models of $\mathrm{ACP^{cr}}$ with condition evaluation and/or generalized condition evaluation are not simply the expansions of the full retrospective splitting bisimulation models $\mathfrak{P}^{\mathrm{cr}}_\kappa$ of $\mathrm{ACP^{cr}}$, for infinite cardinals $\kappa \leq \lambda$, obtained by associating with each operator CE_h and/or GCE_h the corresponding re-labeling operation on conditional transi-

Table 26
Additional axioms for last action conditions ($a \in \mathsf{A}$)

$a \cdot x = a \cdot (\mathcal{J}_a :\rightarrow x)$	J

tion systems with retrospection. As suggested by the structural operational semantics of ACP$^{\mathrm{cr}}$ extended with condition evaluation and generalized condition evaluation, these re-labeling operations have to be adapted in a way similar to the way in which parallel composition had to be adapted to the case with retrospection in Section 16. As mentioned before, full retrospective splitting bisimulation models with process domain $\mathbb{CTS}^{\mathrm{r}}_{\kappa}/\underline{\leftrightarrow}^{\mathrm{r}}$ for $\kappa > \lambda$ are excluded.

Proposition 16, stating that the generalized λ-complete condition evaluation operators supersede the λ-complete condition evaluation operators in the setting of ACP$^{\mathrm{c}}$, goes through in the setting of ACP$^{\mathrm{cr}}$.

Adding state operators to ACP$^{\mathrm{cr}}$ can be done on the same lines as adding generalized evaluation operators to ACP$^{\mathrm{cr}}$, but is more complicated. Roughly speaking, signal emission can be added to ACP$^{\mathrm{cr}}$ in the same way as it is added to ACP$^{\mathrm{c}}$ provided that signals are taken from $\mathcal{C}$. No adaptations like for generalized condition evaluation are needed because signal emission corresponds to condition evaluation that does not persist over performing an action. This property also points at one of the differences between the signal-emission approach to condition evaluation and the other approaches treated in this paper: retrospection has to be resolved in the signal-emission approach before condition evaluation can take place. The case where signals are taken from $\mathcal{C}^{\mathrm{r}}$ is expected to be too complicated to handle.

18 A Variant of ACP$^{\mathrm{cr}}$ with Last Action Conditions

In this section, we outline a process algebra built on ACP$^{\mathrm{cr}}$. It is a variant of ACP$^{\mathrm{cr}}$ in which condition evaluation takes place implicitly. The actions that have been performed in preceding steps determine the result of the implicit condition evaluation. The evaluation mechanism concerned requires a minor adaptation of the axioms of ACP$^{\mathrm{cr}}$.

We take the set $\{\mathcal{J}_a \mid a \in \mathsf{A}\}$ of *last action conditions* as the set of atomic conditions C_{at}. The intuition is that $\mathcal{J}_a$ indicates that action a is performed just now. The retrospection operator now allows for using conditions which express that a certain number of steps ago a certain action must have been performed.

The additional axioms for last action conditions are given in Table 26. More-

Table 27
Axioms adapted to last action conditions ($a, b \in \mathsf{A}_\delta$, $c \in \mathsf{A}$)

$a \cdot x \mid b = (a \mid b) \cdot \Pi^a_0(x)$		CM5J
$a \mid b \cdot x = (a \mid b) \cdot \Pi^b_0(x)$		CM6J
$a \cdot x \mid b \cdot y = (a \mid b) \cdot (\Pi^a_0(x) \parallel \Pi^b_0(y))$		CM7J
$\Pi^+_{>0}(\mathcal{J}_c) = {\sim}\mathcal{J}_c$		RS7Ja
$\Pi^+_{>n+1}(\mathcal{J}_c) = \mathcal{J}_c$		RS7Jb
$\Pi^a_n(b) = b$		LAU1
$\Pi^a_n(b \cdot x) = b \cdot \Pi^a_{n+1}(x)$		LAU2
$\Pi^a_n(x + y) = \Pi^a_n(x) + \Pi^a_n(y)$		LAU3
$\Pi^a_n(\phi :\rightarrow x) = \Pi^a_n(\phi) :\rightarrow \Pi^a_n(x)$		LAU4
$\Pi^a_n(\bot) = \bot$		LAU5
$\Pi^a_n(\top) = \top$		LAU6
$\Pi^a_0(\mathcal{J}_c) = \bot$	if $a \neq c$	LAU7
$\Pi^a_0(\mathcal{J}_c) = \top$	if $a = c$	LAU8
$\Pi^a_{n+1}(\mathcal{J}_c) = \mathcal{J}_c$		LAU9
$\Pi^a_n(-\phi) = -\Pi^a_n(\phi)$		LAU10
$\Pi^a_n(\phi \sqcup \psi) = \Pi^a_n(\phi) \sqcup \Pi^a_n(\psi)$		LAU11
$\Pi^a_n(\phi \sqcap \psi) = \Pi^a_n(\phi) \sqcap \Pi^a_n(\psi)$		LAU12
$\Pi^a_0({\sim}\phi) = {\sim}\phi$		LAU13
$\Pi^a_{n+1}({\sim}\phi) = {\sim}\Pi^a_n(\phi)$		LAU14

over, axioms CM5–CM7 (Table 4) and RS7 (Table 20) must be replaced by axioms CM5J–CM7J, RS7Ja and RS7Jb from Table 27. Axioms CM5–CM7 must be replaced by axioms CM5J–CM7J because, after performing $a \mid b$, it makes no sense to refer back to the actions performed just now by the processes originally following a and b in the process following $a \mid b$. Retrospective conditions in the process originally following a that indicate that a is performed just now should be evaluated to $\top$ and the ones that indicate that another action is performed just now should be evaluated to $\bot$. Retrospective conditions in the process originally following b should be evaluated analogously. This is accomplished by the auxiliary operators $\Pi^a_n : \mathbf{P} \rightarrow \mathbf{P}$ and $\Pi^a_n : \mathbf{C} \rightarrow \mathbf{C}$ (for each $a \in \mathsf{A}_\delta$ and $n \in \mathbb{N}$) of which the defining axioms are axioms LAU1–LAU14 from Table 27. Axiom RS7 must be replaced by axioms RS7Ja and RS7Jb because of the retrospective nature of last action conditions. We mean by this that $\mathcal{J}_a$ can be viewed as a condition of the form ${\sim}\eta$, where η indicates that action a is performed next. We have not introduced corresponding atomic conditions because their use without restrictions would be problematic in alternative composition.

From the axioms of $\mathrm{BPA}_\delta^{\mathrm{cr}}$ and the additional axiom J, we can derive the equation $a \cdot x + b \cdot y = (a + b) \cdot (\mathcal{J}_a :\to x + \mathcal{J}_b :\to y)$. This equation can be used to reduce the number of subprocesses of a process. For example, the equation $a \cdot a' + b \cdot b' = (a + b) \cdot (\mathcal{J}_a :\to a' + \mathcal{J}_b :\to b')$ shows a reduction from 3 subprocesses to 2 subprocesses and the equation $a \cdot (a_1 \cdot a'_1 + a_2 \cdot a'_2) + b \cdot (b_1 \cdot b'_1 + b_2 \cdot b'_2) = (a + b) \cdot (\mathcal{J}_a :\to (a_1 + a_2) \cdot (\mathcal{J}_{a_1} :\to a'_1 + \mathcal{J}_{a_2} :\to a'_2) + \mathcal{J}_b :\to (b_1 + b_2) \cdot (\mathcal{J}_{b_1} :\to b'_1 + \mathcal{J}_{b_2} :\to b'_2))$ shows a reduction from 7 subprocesses to 4 subprocesses.

In order to obtain the full retrospective splitting bisimulation models of the extension of $\mathrm{ACP}^{\mathrm{cr}}$ with last action conditions, retrospective splitting bisimilarity has to be adapted: in the definition of retrospective splitting bisimulation (see Section 14), the two occurrences of $B(s'_1, {\sim}\alpha', s'_2)$ must be replaced by $B(s'_1, {\sim}\alpha' \sqcap \mathcal{J}_a, s'_2)$.

The operators Π_n^a are reminiscent of the operators Π_n^h from Section 17. In fact, if we would exclude full retrospective splitting bisimulation models with process domain $\mathbb{CTS}_\kappa^{\mathrm{r}}/{\underline{\leftrightarrow}}^{\mathrm{r}}$ for κ greater than some infinite cardinal λ, Π_n^a could have been replaced by $\Pi_n^{h_a}$, where $h_a \in \mathcal{H}_\lambda^{\mathrm{r}}$ for $a \in \mathsf{A}$ is defined by $h_a(\mathcal{J}_a) = \top$ and $h_a(\mathcal{J}_b) = \bot$ if $a \neq b$ and $h_\delta \in \mathcal{H}_\lambda^{\mathrm{r}}$ is defined by $h_\delta(\mathcal{J}_a) = \bot$.

19 Concluding Remarks

In this paper, we build on earlier work on ACP. Conditional expressions of the form $\zeta :\to p$ were added to ACP for the first time in [3]. In [4], it was proposed to take the domain of a free Boolean algebra over a given set of generators as the set of conditions. Splitting bisimilarity is based on the variant of bisimilarity that was defined for the first time in [4]. The formulation given here is closer to the one given in [5]. State operators and signal emission were added to ACP for the first time in [22] and [6], respectively. The condition evaluation operators, the generalized evaluation operators and the retrospection operator are new. The variants of splitting bisimilarity, i.e. signal-observing splitting bisimilarity and retrospective splitting bisimilarity, are new as well.

Full bisimulation models were presented in [19] for a first-order extension of ACP. Those models are basically the graph models of ACP, which are most extensively described in [20]. The full splitting bisimilation models of $\mathrm{ACP}^{\mathrm{c}}$ presented in this paper, as well as the full signal-observing splitting bisimilation models of $\mathrm{ACP}^{\mathrm{cs}}$ and the full retrospective splitting bisimilation models of $\mathrm{ACP}^{\mathrm{cr}}$, are adaptations of the full bisimulation models from [19]. The adaptations, in particular for the models of ACP$^{\mathrm{cs}}$ and ACP$^{\mathrm{cr}}$, are substantial.

The above-mentioned variants of full bisimulation models take into account infinitely branching processes, even in the case where the set of atomic conditions (the set of generators) is infinite. We are not aware of previous work presenting models of such generality for a process algebra with conditional expressions. We are also not aware of previous work studying condition evaluation or retrospective conditions in a process algebra with conditional expressions.

In some extensions of ACP with conditional expressions, the conditions are propositions of a three-, four- or five-valued propositional logic, see e.g. [25,26]. It is not clear whether the work presented in this paper can be adapted to those cases, because they bring us outside the domain of Boolean algebras.

In this paper, we give a survey of algebraic theories about processes that include conditional expressions and the main models of those theories. Although our aim is to provide complete axiomatizations, we do not present completeness theorems. We conjecture that the axioms of ACP$^{\mathrm{c}}$, ACP$^{\mathrm{cs}}$ and ACP$^{\mathrm{cr}}$ form complete axiomatizations of the full splitting bisimulation models of ACP$^{\mathrm{c}}$, the full signal-observing splitting bisimilation models of ACP$^{\mathrm{cs}}$ and the full retrospective splitting bisimilation models of ACP$^{\mathrm{cr}}$, respectively, with respect to equations between closed terms; and leave the proofs for future work.

Other options for future work include: development of an extension of ACP$^{\mathrm{cr}}$ with state operators, development of an extension of ACP$^{\mathrm{cr}}$ with signal emission, development of first-order extensions of ACP$^{\mathrm{c}}$, ACP$^{\mathrm{cs}}$ and ACP$^{\mathrm{cr}}$ in the style of [19], and investigations into ways to deal with the history pointers from [8] in the setting of ACP$^{\mathrm{cr}}$.

After the report version of this paper appeared, we have written several closely related papers, namely [27–30]. The first of those, i.e. [27], is essentially an extended abstract of Sections 1–10 of this paper. In [28], it is shown that the threads and services considered in that paper can be viewed as processes that are definable over ACP$^{\mathrm{c}}$. In [29], we present ACP$^{\mathrm{cc}}$, a variant of ACP$^{\mathrm{c}}$ in which the conditions concern the enabledness of actions in the context in which a process is placed. With those conditions, it becomes easy to model preferential choices, which are usually modelled rather indirectly using a priority mechanism. In [30], we add a constant for a process that is only capable of terminating successfully to ACP$^{\mathrm{c}}$, ACP$^{\mathrm{cs}}$ and ACP$^{\mathrm{cr}}$. It happens that the addition of this constant, often referred to as the empty process constant, is unproblematic. Therefore, we look upon [30] primarily as supplementary material to this paper.

Acknowledgements

We thank an anonymous referee for his/her valuable comments.

References

[1] J. A. Bergstra, J. W. Klop, Process algebra for synchronous communication, Information and Control 60 (1/3) (1984) 109–137.

[2] J. C. M. Baeten, W. P. Weijland, Process Algebra, Vol. 18 of Cambridge Tracts in Theoretical Computer Science, Cambridge University Press, Cambridge, 1990.

[3] J. C. M. Baeten, J. A. Bergstra, S. Mauw, G. J. Veltink, A process specification formalism based on static COLD, in: J. A. Bergstra, L. M. G. Feijs (Eds.), Algebraic Methods II: Theory, Tools and Applications, Vol. 490 of Lecture Notes in Computer Science, Springer-Verlag, 1991, pp. 303–335.

[4] J. C. M. Baeten, J. A. Bergstra, Process algebra with signals and conditions, in: M. Broy (Ed.), Programming and Mathematical Methods, Vol. F88 of NATO ASI Series, Springer-Verlag, 1992, pp. 273–323.

[5] J. A. Bergstra, A. Ponse, J. J. van Wamel, Process algebra with backtracking, in: J. W. de Bakker, W. P. de Roever, G. Rozenberg (Eds.), A Decade of Concurrency (Reflections and Perspectives), Vol. 803 of Lecture Notes in Computer Science, Springer-Verlag, 1994, pp. 46–91.

[6] J. C. M. Baeten, J. A. Bergstra, Process algebra with propositional signals, Theoretical Computer Science 177 (1997) 381–405.

[7] L. Aceto, W. J. Fokkink, C. Verhoef, Structural operational semantics, in: J. A. Bergstra, A. Ponse, S. A. Smolka (Eds.), Handbook of Process Algebra, Elsevier, Amsterdam, 2001, pp. 197–292.

[8] J. C. M. Baeten, J. A. Bergstra, Deadlock behaviour in split and ST bisimulation, in: I. Castellani, C. Palamidessi (Eds.), EXPRESS'98, Vol. 16 of Electronic Notes in Theoretical Computer Science, Elsevier, 1998, pp. 101–114.

[9] M. Hennessy, R. Milner, Algebraic laws for non-determinism and concurrency, Journal of the ACM 32 (1985) 137–161.

[10] S. D. Brookes, C. A. R. Hoare, A. W. Roscoe, A theory of communicating sequential processes, Journal of the ACM 31 (3) (1984) 560–599.

[11] J. D. Monk, R. Bonnet (Eds.), Handbook of Boolean Algebras, Vol. 1, Elsevier, Amsterdam, 1989.

[12] J. R. Shoenfield, Mathematical Logic, Addison-Wesley Series in Logic, Addison-Wesley, Reading, MA, 1967.

[13] C. C. Chang, H. J. Keisler, Model Theory, 3rd Edition, Vol. 73 of Studies in Logic and the Foundations of Mathematics, Elsevier, Amsterdam, 1990.

[14] C. A. R. Hoare, I. J. Hayes, He Jifeng, C. C. Morgan, A. W. Roscoe, J. W. Sanders, I. H. Sorensen, J. M. Spivey, B. A. Sufrin, Laws of programming, Communications of the ACM 30 (1987) 672–686.

[15] P. R. Halmos, Lectures on Boolean Algebras, Mathematical Studies, Van Nostrand, Princeton, NJ, 1963.

[16] R. Sikorski, Boolean Algebras, 3rd Edition, Springer-Verlag, Berlin, 1969.

[17] N. Busi, R. J. van Glabbeek, R. Gorrieri, Axiomatising ST-bisimulation semantics, in: E.-R. Olderog (Ed.), PROCOMET'94, Vol. 56 of IFIP Transactions A, North-Holland, 1994, pp. 169–188.

[18] K. G. Larsen, A. Skou, Bisimulation through probabilistic testing, Information and Computation 94 (1) (1991) 1–28.

[19] J. A. Bergstra, C. A. Middelburg, Model theory for process algebra, in: A. Middeldorp, V. van Oostrom, F. van Raamsdonk, R. C. de Vrijer (Eds.), Processes, Terms and Cycles: Steps on the Road to Infinity, Vol. 3838 of Lecture Notes in Computer Science, Springer-Verlag, 2005, pp. 445–495.

[20] J. C. M. Baeten, J. A. Bergstra, J. W. Klop, On the consistency of Koomen's fair abstraction rule, Theoretical Computer Science 51 (1/2) (1987) 129–176.

[21] J. F. Groote, A. Ponse, Proof theory for μCRL: A language for processes with data, in: D. J. Andrews, J. F. Groote, C. A. Middelburg (Eds.), Semantics of Specification Languages, Workshops in Computing Series, Springer-Verlag, 1994, pp. 232–251.

[22] J. C. M. Baeten, J. A. Bergstra, Global renaming operators in concrete process algebra, Information and Control 78 (3) (1988) 205–245.

[23] J. A. Bergstra, C. A. Middelburg, Y. S. Usenko, Discrete time process algebra and the semantics of SDL, in: J. A. Bergstra, A. Ponse, S. A. Smolka (Eds.), Handbook of Process Algebra, Elsevier, Amsterdam, 2001, pp. 1209–1268.

[24] J. A. Bergstra, C. A. Middelburg, Process algebra for hybrid systems, Theoretical Computer Science 335 (2/3) (2005) 215–280.

[25] J. A. Bergstra, A. Ponse, Process algebra and conditional composition, Information Processing Letters 80 (2001) 41–49.

[26] M. B. van der Zwaag, Models and logics for process algebra, Ph.D. thesis, Programming Research Group, University of Amsterdam, Amsterdam (2002).

[27] J. A. Bergstra, C. A. Middelburg, Strong splitting bisimulation equivalence, in: J. L. Fiadeiro, N. Harman, M. Roggenbach, J. Rutten (Eds.), CALCO 2005, Vol. 3629 of Lecture Notes in Computer Science, Springer-Verlag, 2005, pp. 83–97.

[28] J. A. Bergstra, C. A. Middelburg, Thread algebra with multi-level strategies, Fundamenta Informaticae 71 (2/3) (2006) 153–182.

[29] J. A. Bergstra, C. A. Middelburg, Preferential choice and coordination conditions, to appear in *Journal of Logic and Algebraic Programming.* Preliminary version: Computer Science Report 05-14, Department of Mathematics and Computer Science, Eindhoven University of Technology (April 2005).

[30] J. A. Bergstra, C. A. Middelburg, Process algebra with conditionals in the presence of epsilon, Computer Science Report 05-15, Department of Mathematics and Computer Science, Eindhoven University of Technology (May 2005).